Statistique de l'assurance

Arthur Charpentier

Arthur Charpentier

Université de Rennes 1
Université de Montréal
2010-2011

Statistique de l'assurance, STT 6705V
Statistique de l'assurance II

partie 1 - assurance non-vie
tarification & provisionnement

http ://freakonometrics.blog.free.fr/

Table des matières

1 La tarification a priori **5**
1.1 Les modèles linéaires généralisés 7
1.1.1 Le cadre général des GLM 7
1.1.2 Approche économétrique de la tarification 9
1.1.3 Estimation des paramètres 10
1.1.4 Interprétation d'une régression 13
1.1.5 Extension à d'autres familles de lois 16
1.1.6 De la qualité d'une régression 16
1.1.7 Les variables tarifaires continues et la nonlinéarité . . 19
1.1.8 Les modèles nonlinéaires multivariés 23
1.2 Modéliser des variables indicatrices 24
1.2.1 La régression logistique ou probit 24
1.2.2 Les arbres de régression 26
1.2.3 Probabilité d'avoir (au moins) un sinistre dans l'année 30
1.2.4 Probabilité d'avoir un gros sinistre dans l'année 31
1.3 Modéliser la fréquence de sinistralité 34
1.3.1 Un peu d'analyse descriptive 34
1.3.2 La méthode des marges 38
1.3.3 Prise en compte de l'exposition et variable offset . . . 40
1.3.4 Prise en compte de la surdispersion 41
1.3.5 Les modèles *zero-inflated* 44
1.3.6 Régression simple versus régression multiple 47
1.3.7 Prédiction de la fréquence par police 47
1.4 Modéliser les coûts individuels des sinistres 50
1.4.1 Modèle Gamma et modèle lognormal 50
1.4.2 Modélisation des grands sinistres 57
1.4.3 Ecrêtement des grands sinistres 58
1.5 Modéliser les coûts par police 60
1.5.1 Les modèles Tweedie comme modèle Poisson composé 60

2 Les provisions pour sinistres à payer **63**
2.1 La problématique du provisionnment 63

2.1.1 Quelques définitions et notations, aspects règlementaires et comptables . . . 63
2.1.2 Formalisation du problème du provisionnement . . . 65
2.2 Les cadences de paiements et la méthode Chain Ladder . . . 66
2.3 De Mack à Merz & Wüthrich . . . 69
2.3.1 Quantifier l'incertitude dans une prédiction . . . 69
2.3.2 Le formalisme de Mack . . . 70
2.3.3 La notion de *tail factor* . . . 72
2.3.4 Des estimateurs des paramètres à l'incertitude sur le montant des provisions . . . 72
2.3.5 Un mot sur Munich-Chain Ladder . . . 73
2.3.6 L'incertitude à un an de Merz & Wüthrich . . . 78
2.4 Régression Poissonnienne et approches économétriques . . . 83
2.4.1 Les modèles à facteurs, un introduction historique . . . 83
2.4.2 Les modèles de de Vylder et de Chritophides . . . 83
2.4.3 La régression poissonnienne de Hachemeister & Stanard 85
2.4.4 Incertitude dans un modèle de régression . . . 87
2.4.5 Le modèle binomial-négative . . . 91
2.4.6 Quel modèle de régression ? . . . 91
2.5 Les triangles multivariés . . . 92
2.5.1 Hypohtèse d'indépendance entre les triangles, et lois paramétriques . . . 93
2.5.2 Le modèle de Mack bivarié . . . 95
2.5.3 Modèles économétriques pour des risques multiples . . . 96
2.6 Borhutter-Fergusson, Benktander et les méthodes bayésiennes 97
2.6.1 Le modèle de Borhutter-Ferguson et l'introduction d'un avis d'expert . . . 97
2.6.2 Benktander . . . 98
2.6.3 La méthode dite *Cape-Code* . . . 99
2.6.4 Les approches Bayésiennes . . . 100
2.6.5 Approche bayésienne sur les facteurs de développement 104

Chapitre 1

La tarification a priori

Pour chaque police d'assurance, la prime est fonction de variables dites de tarification. Généralement, on considère

- des informations sur l'assuré, comme l'âge ou le sexe pour un particulier, ou le secteur d'activité et le nombre de salariés pour une entreprise,
- des informations sur le bien assuré, comme l'âge du véhicule, la puissance ou la marque en assurance auto, la surface du logement en multirisque habitation, le chiffre d'affaire de l'entreprise en perte d'exploitation,
- des informations géograhiques comme le revenu moyen dans la commune ou le département, la densité de population, etc.

La *fréquence* est le nombre de sinistres divisé par l'exposition (correspondant au nombre d'années police) pour une police d'assurance, ou un groupe de polices d'assurance. La plupart des contrats étant annuels, on ramènera toujours le nombre de sinistres à une exposition annuelle lors du calcul de la prime, et on notera N la variable aléatoire associée. Durant la période d'exposition, on notera Y_i les coûts des sinistres, c'est à dire les indemnités versées par l'assureur à l'assuré (ou une tierce personne). La charge totale par police est alors $S = 0$ s'il n'y a pas eu de sinistres, ou sinon :

$$S = Y_1 + \cdots + Y_N = \sum_{i=1}^{N} Y_i.$$

Classiquement (et ce point sera important pour constituer la base de données) $Y_i > 0$ et N est alors le nombre de sinistres en excluant les sinistres classés sans suite (i.e. de coût nul).

La prime pure est $\mathbb{E}(S) = \mathbb{E}(N) \cdot \mathbb{E}(Y_i)$ dès lors que les coûts individuels sont i.i.d., indépendants du nombre de sinistres. Dans le cas où la fréquence et les charges sont hétérogènes, l'hétérogénéité étant caractérisée par une information Ω, la prime pure devrait être :

$$\mathbb{E}(S|\Omega) = \mathbb{E}(N|\Omega) \cdot \mathbb{E}(Y_i|\Omega).$$

Le facteur d'hétérogénéité Ω étant inconnu, on utilise les variables tarifaires à notre disposition pour obtenir un proxi de ces espérances conditionnelles. On cherche alors $\boldsymbol{X} = (X_1, \cdots, X_k)$ un ensemble de variables explicatives telles que

$$\mathbb{E}(S|\boldsymbol{X}) = \mathbb{E}(N|\boldsymbol{X}) \cdot \mathbb{E}(Y_i|\boldsymbol{X}).$$

Pour importer les bases de données, on utilise le code suivant (seuls les sinistres de responsabilité civile nous intéressent),

```
 sinistreUdM <- read.table("http://perso.univ-rennes1.fr/arthur.charpentier/sinis
+ header=TRUE,sep=";")
> sinistres=sinistreUdM[sinistreUdM$garantie=="1RC",]
> contratUdM <- read.table("http://perso.univ-rennes1.fr/arthur.charpentier/contr
+ header=TRUE,sep=";")
```

Pour consituer une base contenant les nombres de sinistres, le code est le suivant :

```
> T=table(sinistres$nocontrat)
> T1=as.numeric(names(T))
> T2=as.numeric(T)
> nombre1 = data.frame(nocontrat=T1,nbre=T2)
> I = contratUdM$nocontrat%in%T1
> T1=contratUdM$nocontrat[I==FALSE]
> nombre2 = data.frame(nocontrat=T1,nbre=0)
> nombre=rbind(nombre1,nombre2)
> base = merge(contratUdM,nombre)
> head(base)
  nocontrat exposition zone puissance agevehicule ageconducteur bonus
1        27       0.87    C         7           0            56    50
2       115       0.72    D         5           0            45    50
3       121       0.05    C         6           0            37    55
4       142       0.90    C        10          10            42    50
5       155       0.12    C         7           0            59    50
6       186       0.83    C         5           0            75    50
  marque carburant densite region nbre
1     12         D      93     13    0
2     12         E      54     13    0
3     12         D      11     13    0
4     12         D      93     13    0
5     12         E      73     13    0
6     12         E      42     13    0
```

La base `nombre` contient, par police, le nombre de sinistres en responsabilité civile déclaré par l'assuré pendant l'année d'observation. Parmi les variables d'intérêt,

- `densite` est la densité de population dans la commune où habite le conducteur principal,
- `zone` : zone A B C D E ou F, selon la densité en nombre d'habitants par km^2 de la commune de résidence
- `marque` : marque du véhicule selon la table suivante (1 Renault Nissan ; 2 Peugeot Citroën ; 3 Volkswagen Audi Skoda Seat ; 4 Opel GM ; 5 Ford ; 6 Fiat ; 10 Mercedes Chrysler ; 11 BMW Mini ;12 Autres japonaises et coréennes ; 13 Autres européennes ; 14 Autres marques et marques inconnues)
- `region` : code à 2 chiffres donnant les 22 régions françaises (code INSEE)
- `ageconducteur` : âge du conducteur principal en début de la couverture,
- `agevehicule` : âge du véhicule en début de période.

Nous disposons aussi d'un numéro de police `no` permettant de fusionner les deux bases, et donc d'associer à la charge d'un sinistre les caractéristiques du conducteur et du véhicule.

1.1 Les modèles linéaires généralisés

Depuis quelques années, l'outil principal utilisé en tarification est le modèle linéaire généralisé, développé par [22], et dont la mise en oeuvre en assurance est détaillée dans [17], [7], [6], [25] ou [9]. Dans cette section, nous allons présenter le cadre des GLM, ainsi que leur mise en oeuvre sous `R`, avant de rentrer dans l'application en tarification dans les sections suivantes.

1.1.1 Le cadre général des GLM

Les modèles linéaires généralisés sont une généralisation du modèle linéaire Gaussien, obtenu en autorisant d'autres lois (conditionnelles) que la loi Gaussienne. Les lois possibles doivent appartenir à la famille exponentielle, i.e. dont la densité (ou mesure de probabilité dans le cas discret) s'écrit :

$$f(y|\theta,\phi) = \exp\left(\frac{y\theta - b(\theta)}{\phi} + c(y,\phi)\right)$$

Exemple 1.1 *La* loi normale $\mathcal{N}(\mu,\sigma^2)$ *appartient à cette famille, avec* $\theta = \mu$, $\phi = \sigma^2$, $b(\theta) = \theta^2/2$ *et*

$$c(y,\phi) = -\frac{1}{2}\left(\frac{y^2}{\sigma^2} + \log(2\pi\sigma^2)\right), \quad y \in \mathbb{R},$$

Exemple 1.2 *La* loi de Poisson $\mathcal{P}(\lambda)$ *appartient à cette famille,*

$$f(y|\lambda) = \exp(-\lambda)\frac{\lambda^y}{y!} = \exp\Big(y\log\lambda - \lambda - \log y!\Big), \quad y \in \mathbb{N},$$

avec $\theta = \log\lambda$, $\phi = 1$, $b(\theta) = \exp\theta = \lambda$ *et* $c(y,\phi) = -\log y!$.

Exemple 1.3 *La* loi binomiale $\mathcal{B}(n,p)$ *correspond au cas* $\theta = \log\{p/(1-p)\}$, $b(\theta) = n\log(1+\exp(\theta))$, $\phi = 1$ *et* $c(zy,\phi) = \log\binom{n}{y}$.

Exemple 1.4 *La* loi Gamma *est également dans la famille exponentielle,*

$$f(y|\mu,\nu) = \frac{1}{\Gamma(\nu)}\left(\frac{\nu}{\mu}\right)^{\nu} y^{\nu-1}\exp\left(-\frac{\nu}{\mu}y\right),\ \ y\in\mathbb{R}_+,$$

avec $\theta = -\dfrac{1}{\mu}$, $b(\theta) = -\log(-\theta)$ *et* $\phi = \nu^{-1}$.

Pour une variable aléatoire Y dont la densité est de la forme exponentielle, alors

$$\mathbb{E}(Y) = b'(\theta) \text{ et } \mathrm{Var}(Y) = b''(\theta)\phi$$

de telle sorte que la variance de Y apparaît comme le produit de deux fonctions,

- la première, $b''(\theta)$, qui dépend uniquement du paramètre θ est appelée *fonction variance*
- la seconde est indépendante de θ et dépend uniquement de ϕ

En notant $\mu = \mathbb{E}(Y)$, on voit que le paramètre θ est lié à la moyenne μ. La fonction variance peut donc être définie en fonction de μ , nous la noterons dorénavant $V(\mu)$.

Exemple 1.5 *Dans le cas de la loi normale,* $V(\mu) = 1$, *dans le cas de la loi de Poisson,* $V(\mu) = \mu$ *alors que dans le cas de la loi Gamma,* $V(\mu) = \mu^2$.

Notons que la fonction variance caractérise complètement la loi de la famille exponentielle. Chacune des lois de la famille exponentielle possède une fonction de lien spécifique, dite *fonction de lien canonique*, permettant de relier l'espérance μ au paramètre naturel θ. Le lien canonique est tel que $g_\star(\mu) = \theta$. Or, $\mu = b'(\theta)$ donc $g_\star(\cdot) = b'(\cdot)^{-1}$.

Exemple 1.6 *Dans le cas de la loi normale,* $\theta = \mu$ *(*`link='identity'`*), dans le cas de la loi de Poisson,* $\theta = \log(\mu)$ *(*`link='log'`*) alors que dans le cas de la loi Gamma,* $\theta = 1/\mu$ *(*`link='inverse'`*).*

Sous `R`, la syntaxe des modèles linéaires généralisées est :

```
> glm(Y~X1+X2+X3+offset(Z), family =quasipoisson(link='log'),
+ data, weights)
```

ce qui correspond à un modèle

$$\mathbb{E}(Y_i|\boldsymbol{X}_i) = \mu_i = g^{-1}\left(\boldsymbol{X}_i'\boldsymbol{\beta} + \xi_i\right) \text{ et } \text{Var}(Y_i|\boldsymbol{X}_i) = \frac{\phi V(\mu_i)}{\omega_i}$$

où `Y` est le vecteur des Y_i que l'on cherche à modéliser (le nombre de sinistres de la police i par exemple), `X1`, `X2` et `X3` sont les variables explicatives qui peuvent être qualitatives (on parlera de facteurs) ou quantitatives, `link='log'` indique que g est la fonction log, `family=poisson` revient à choisir une fonction variance V identité, alors que `family=quasipoisson` revient à choisir une fonction variance V identité avec un paramètre de dispersion ϕ à estimer, `offset` correspond à la variable ξ_i, et `weights` le vecteur ω_i. Cette fonction `glm` calcule alors des estimateurs de $\boldsymbol{\beta}$ et ϕ, entre autres, car comme pour le modèle linéaire gaussien (la fonction `lm`) on peut obtenir des prédictions, des erreurs, ainsi qu'un grand nombre d'indicateurs relatifs à la qualité de l'ajustement.

1.1.2 Approche économétrique de la tarification

Cette famille de lois (dite *exponentielle*) va s'avérer être particulièrement utile pour construire des modèles économétriques beaucoup plus généraux que le modèle Gaussien usuel. On suppose disposer d'un échantillon $(Y_i, \boldsymbol{X}_i)$, où les variables $\boldsymbol{X}_i$ sont des informations exogènes sur l'assuré ou sur le bien assuré, et où Y_i est la variable d'intérêt, qui sera

- une variable booléenne 0/1, par exemple l'assuré i a-t-il été victime d'un accident l'an dernier,
- une variable de comptage, à valeurs dans $\boldsymbol{N}$, par exemple le nombre d'accident de l'assuré i l'an passé,
- une variable positive, à valeurs dans $\boldsymbol{R}^+$, par exemple le coût du sinistre i, ou bien la durée entre la survenance et la déclaration du sinistre.

On supposera que, conditionnellement aux variables explicatives $\boldsymbol{X}$, les variables Y sont indépendantes, et identiquement distribuées. En particulier, on partira d'un modèle de la forme

$$f(y_i|\theta_i, \phi) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi)\right)$$

où l'on supposera que

$$g(\mu_i) = \eta_i = \boldsymbol{X}_i'$$

pour une fonction de lien $g(\cdot)$ donnée (on gardera ainsi un score *linéaire* en les variables explicatives), et où, pour rappel,

$$\mu_i = \mathbb{E}(Y_i|\boldsymbol{X}_i)$$

La fonction lien est la fonction qui permet de lier les variables explicatives $\boldsymbol{X}$ à la prédiction μ, alors que la loi apparaît via la fonction variance, sur

la forme de l'hétéroscédasticité et l'incertitude associée à la prédiction. Le petit exemple ci-dessous permet de visualiser sur un petit de données simple six régressions GLM différentes,

```
> x <- c(1,2,3,4,5)
> y <- c(1,2,4,2,6)
> base <- data.frame(x,y)
> plot(x,y,pch=19,cex=1.5)
> regNId  <- glm(y~x,family=gaussian(link="identity"))
> regNlog <- glm(y~x,family=gaussian(link="log"))
> regPId  <- glm(y~x,family=poisson(link="identity"))
> regPlog <- glm(y~x,family=poisson(link="log"))
> regGId  <- glm(y~x,family=Gamma(link="identity"))
> regGlog <- glm(y~x,family=Gamma(link="log"))
```

La prédiction (ainsi qu'un intervalle de confiance) pour chacun de ces modèles est présentée sur la Figure 1.1. Le code de base pour obtenir la prédiction avec un intervalle de confiance (à 95%) est simplement

```
> plot(x,y,pch=19,cex=1.5)
> abs <- seq(0,7,by=.1)
> yp <- predict(regNId,newdata=data.frame(x=abs),se.fit = TRUE,
+ type="response")
> lines(abs,yp$fit,lwd=2)
> lines(abs,yp$fit+2*yp$se.fit,lty=2)
> lines(abs,yp$fit-2*yp$se.fit,lty=2)
```

Remarque 1.1 *De la même manière qu'en économétrie linéaire, il est aussi possible d'allouer des poids à chacune des observations ω_i. Mais nous n'en parlerons pas trop ici. Il peut s'agir de pondération décroisantes avec le temps, attribuées à des années trop anciennes, si l'on utilise des données sur une période plus longue, par exemple.*

1.1.3 Estimation des paramètres

La loi de Y sachant $\boldsymbol{X}$ étant spécifiée, on obtient numériquement les estimateurs de $\boldsymbol{\beta}$ et ϕ par maximisation de la vraisemblance.

```
> logv=function(beta){
+ L=beta[1]+beta[2]*sinistres$ageconducteur
+ -sum(log(dpois(sinistres$nombre,exp(L))))
+ }
> nlm(f = logv, p = beta)
$minimum
```

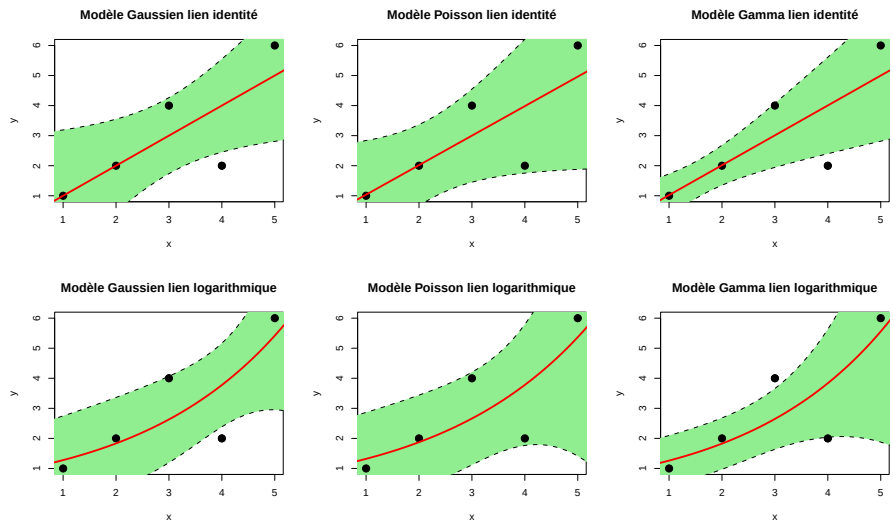


FIGURE 1.1 – Prédiction par 6 modèles linéaires différents, 3 lois et 2 fonctions de lien, avec les intervalles de confiance de prédiction.

```
[1] 113429.7

$estimate
[1] -3.157159895 -0.001900410

$gradient
[1] 0.01069032 1.31089786

$code
[1] 3

$iterations
[1] 25
>  glm(nombre~ageconducteur,data=sinistres,family="poisson")$coefficients
  (Intercept) ageconducteur
 -3.157198596  -0.001899561
```

Notons qu'il est aussi possible d'utiliser une régression linéaire pondérée. En effet, on cherche à maximiser ici une (log)-vraisemblance (ou une déviance comme nous le verrons plus tard), qui s'écrit dans le cas des modèles exponentiels,

$$\log \mathcal{L} = \sum_{i=1}^{n} \left[\frac{Y_i \theta_i - b(\theta_i)}{a(\psi)} - c(Y_i, \psi) \right]$$

mais comme on cherche les paramètres β, on note que le maximum de vraisemblance du paramètre β est atteint au même point que le maximum de la fonction

$$\log \mathcal{L} = \sum_{i=1}^{n} [Y_i \theta_i - b(\theta_i)]$$

Le maximum est alors atteint en $\widehat{\beta}$ tel que

$$\frac{\partial}{\partial \boldsymbol{\beta}} \log \mathcal{L} == \sum_{i=1}^{n} [Y_i - b'(\theta_i)] \frac{\partial}{\partial \theta_i \boldsymbol{\beta}} = 0.$$

Or $\mu_i = g(\eta_i) = g(\boldsymbol{X}_i' \boldsymbol{\beta}) = b'(\theta_i)$, et donc

$$b'(\theta_i)] \frac{\partial}{\partial \theta_i \boldsymbol{\beta}} = g(\boldsymbol{X}_i' \boldsymbol{\beta}) X_i$$

On cherche alors à résoudre

$$\sum_{i=1}^{n} [Y_i - \mu_i] \frac{g'(\boldsymbol{X}_i' \boldsymbol{\beta})}{\mathrm{V}(\mu_i)} \boldsymbol{X}_i,$$

Ce qui correspondrait à la condition du premier ordre dans une régression pondérée, où la matrice de poids serait $W = [w_{i,j}]$, où $w_{i,j} = 0$ si $i \neq j$, et sinon

$$w_{i,i} = \frac{1}{\mathrm{Var}(Y_i)} = \frac{1}{\mu_i} = \frac{1}{g^{-1}(\boldsymbol{X}_i' \boldsymbol{\beta})}$$

Mais cette matrice de poids étant inconnue (elle dépend des paramètres que l'on cherche à estimer), on met en place une itération de régression pondérée, la matrice de poids étant calculée à partir des coefficients de l'étape précédante.

Dans le cas d'une régression log-Poisson, le code devient,

```
>  BETA=matrix(NA,101,2)
>  REG=lm(nombre~ageconducteur,data=sinistres)
>  beta=REG$coefficients
>  BETA[1,]=beta
>  for(i in 2:15){
+  eta=beta[1]+beta[2]*sinistres$ageconducteur
+  mu=exp(eta)
+  w=mu
+  z=eta+(sinistres$nombre-mu)/mu
+  REG=lm(z~sinistres$ageconducteur,weights=w)
+  beta=REG$coefficients
+  BETA[i,]=beta
+  }
```

```
> BETA
              [,1]          [,2]
  [1,]  0.04239008 -7.371466e-05
  [2,] -0.91696821 -1.418714e-04
  [3,] -1.81086687 -3.136888e-04
  [4,] -2.55133907 -6.958340e-04
  [5,] -3.00654605 -1.315441e-03
  [6,] -3.14670636 -1.803882e-03
  [7,] -3.15715335 -1.898126e-03
  [8,] -3.15719860 -1.899561e-03
  [9,] -3.15719860 -1.899561e-03
 [10,] -3.15719860 -1.899561e-03
 [11,] -3.15719860 -1.899561e-03
 [12,] -3.15719860 -1.899561e-03
 [13,] -3.15719860 -1.899561e-03
 [14,] -3.15719860 -1.899561e-03
 [15,] -3.15719860 -1.899561e-03
```

qui converge très rapidement (vers les bonnes valeurs).

1.1.4 Interprétation d'une régression

Considérons tout simplement une régression de la fréquence annuelle de sinistre sur l'âge du conducteur. On supposera un modèle Poissonnien.

```
> reg1 <- glm(nombre~ageconducteur,data=nombre,family=poisson(link="log"),
+ offset=log(exposition))
> summary(reg1)

Call:
glm(formula = nombre ~ ageconducteur, family = poisson(link = "log"),
    data = nombre, offset = log(exposition))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.5685  -0.3527  -0.2611  -0.1418  13.3247

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.1369116  0.0207723 -102.87   <2e-16 ***
ageconducteur -0.0101679  0.0004397  -23.12   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 171919  on 678012  degrees of freedom
Residual deviance: 171373  on 678011  degrees of freedom
AIC: 222190

Number of Fisher Scoring iterations: 6
```

Avec un lien logarithmique, le modèle est multplicatif. Le multiplicateur est ici

```
> exp(coefficients(reg1)[2])
ageconducteur
    0.9898836
```

Autrement dit, tous les ans, la probabilité d'avoir un accident diminue de $1 - 0.9898 = 1.011\%$.

Si l'on considère des classes d'âges (définies *a priori*, nous reviendrons par la suite sur la construction *optimale* des classes), on obtient la régression suivante :

```
> seuils = c(17,21,25,30,45,55,65,80,120)
> nombre$agecut <- cut(nombre$ageconducteur,breaks=seuils)
> reg2 <- glm(nombre~agecut ,data=nombre,family=poisson(link="log"),
+ offset=log(exposition))
> summary(reg2)

Call:
glm(formula = nombre ~ agecut, family = poisson(link = "log"),
    data = nombre, offset = log(exposition))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.6566  -0.3522  -0.2601  -0.1413  13.2465

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -1.55416    0.03277  -47.42   <2e-16 ***
agecut(21,25]  -0.52724    0.04186  -12.60   <2e-16 ***
agecut(25,30]  -0.95181    0.03865  -24.62   <2e-16 ***
agecut(30,45]  -1.08673    0.03441  -31.58   <2e-16 ***
agecut(45,55]  -1.04649    0.03500  -29.90   <2e-16 ***
agecut(55,65]  -1.19279    0.03709  -32.16   <2e-16 ***
agecut(65,80]  -1.27536    0.03876  -32.90   <2e-16 ***
agecut(80,120] -1.24017    0.06743  -18.39   <2e-16 ***
```

```
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 171919  on 678012  degrees of freedom
Residual deviance: 170589  on 678005  degrees of freedom
AIC: 221417

Number of Fisher Scoring iterations: 6
```

Notons qu'il est aussi possible de taper directement

```
> reg2 = glm(nombre~cut(ageconducteur,breaks=seuils),data=nombre,
+ family=poisson(link="log"),offset=log(exposition))
```

La classe de référence est ici celle des jeunes conducteurs `(17,21]`. Relativement à cette classe, on note que toutes les classes ont une probabilité d'avoir un accident plus faible. Pour un conducteur de la classe `(30,45]`, on note qu'il a 66% de chances en moins d'avoir un accident dans l'année qu'un jeune conducteur,

```
> exp(coefficients(reg2)[4])
cut(ageconducteur, breaks = seuils)(30,45]
                                 0.3373169
```

Au lieu de comparer à la classe des jeunes conducteurs, on peut aussi comparer au conducteur moyen.

```
> seuils = c(17,21,25,30,45,55,65,80,120)
> reg2 = glm(nombre~0+cut(ageconducteur,breaks=seuils),
+ data=nombre,family=poisson(link="log"),offset=log(exposition))
```

Les multiplicateurs sont alors

```
> reg2b <- glm(nombre~1,data=nombre,family=poisson(link="log"),
+ offset=log(exposition))
> moyenne <- exp(coefficients(reg2b))
> reg2c <- glm(nombre~0+cut(ageconducteur,breaks=seuils),
+ data=nombre,family=poisson(link="log"),offset=log(exposition))
> exp(coefficients(reg2c))/moyenne
```

Une personne de la classe `(17,21]` a ainsi 2.86 fois plus de chance que l'assuré moyen d'avoir un accident.

1.1.5 Extension à d'autres familles de lois

Les modèles linéaires généralisés ont été définis pour des lois (de Y, conditionnelles aux variables explicatives $\boldsymbol{X}$) appartenant à la famille exponentielle. Il est toutefois possible de généraliser. Les lois de `library(gamlss)` sont des lois à quatre paramètres, (μ, σ, ν, τ), où μ est un paramètre de localisation (e.g. la moyenne), σ un paramètre d'échelle (e.g. l'écart-type), et où ν et τ sont des paramètres d'asymétrie et d'épaisseur de queue (e.g. la skewness et la kurtosis). Ces quatre paramètres peuvent être fonction des variables explicatives au travers d'une fonction de lien,

$$\begin{cases} \mu = g_\mu^{-1}(\boldsymbol{X}\boldsymbol{\alpha}) \\ \sigma = g_\sigma^{-1}(\boldsymbol{X}\boldsymbol{\beta}) \\ \nu = g_\nu^{-1}(\boldsymbol{X}\boldsymbol{\gamma}) \\ \tau = g_\tau^{-1}(\boldsymbol{X}\boldsymbol{\delta}) \end{cases}$$

Parmi les lois classiques, on retrouvera celles données dans la Table 1.1.

loi	R	μ	σ	ν	τ
Binomiale	`BI`	logit	-	-	-
Normale	`NO`	identité	log	-	-
Poisson	`PO`	log	-	-	-
Gamma	`GA`	logit	-	-	-
inverse Gaussienne	`IG`	log	log	-	-
Gumbel	`GU`	identité	log	-	-
lognormale	`LNO`	log	log	-	-
binomiale négative (Poisson-Gamma)	`NBI`	log	log	-	-
Poisson-inverse Gaussien	`PIG`	log	log	-	-
Weibull	`WEI`	log	log	-	-
zero inflated Poisson	`ZIP`	log	logit	-	-

TABLE 1.1 – Les différentes lois et modèles de `library(gamlss)@`.

Dans sa version la plus simple, on retrouve le modèle proposé par [11],

$$\begin{cases} Y_i = \boldsymbol{X}_i'\boldsymbol{\beta} + \varepsilon_i, \text{modèle en moyenne} \\ \log \varepsilon_i^2 = \boldsymbol{Z}_i'\boldsymbol{\alpha} + u_i, \text{modèle en variance} \end{cases}$$

où u_i est un bruit i.i.d. suivant une loi Gamma. Cette fonction particulière est obtenue à l'aide de la fonction `lm.disp` de `library(dispmod)`.

1.1.6 De la qualité d'une régression

Pour mesurer les performances d'une régression, ou plus généralement d'un modèle quel qu'il soit, il faut se donner une *fonction de risque* $R(\cdot, \cdot)$ qui mesure la distance entre Y et sa prédiction $\widehat{Y}$. Classiquement, on utilise

la norme L^2, correspond à l'erreur quadratique $R(Y,\widehat{Y}) = [Y - \widehat{Y}]^2$ ou la norme L^1 , correspondant à l'erreur absolue $R(Y,\widehat{Y}) = |Y - \widehat{Y}|$.

Si on reprend l'exemple de la section 1.1.2, les résidus sont représenté sur la Figure 1.2. Les résidus de gauche sont les résidus bruts, c'est à dire la différence entre Y_i et $\widehat{Y}_i$. A droite, ce sont les résidus de Pearson, i.e.

$$\widehat{\varepsilon}_i = \frac{Y_i - \widehat{Y}_i}{\sqrt{V(\widehat{Y}_i)}}$$

où V est la fonction variance.

```
> RNIr <- residuals(regNId,type="response")
> RNIp <- residuals(regNId,type="pearson")
```

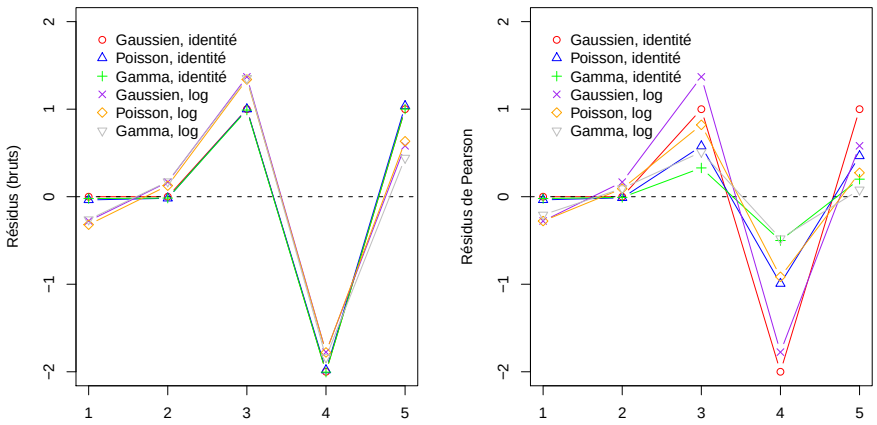


FIGURE 1.2 – Résidus de la régression.

Les résidus de Pearson permettent de prendre en compte de l'hétéroscédasticité qui apparaîtra dès lors que l'on quite le modèle Gaussien (la fonction variance ne sera alors plus constante). Pour le modèle log-Poisson, les erreurs L^1 et L^2 sont respectivement

```
>   cat("Erreur L1 =",sum(abs(RPL))
Erreur L1 = 4.196891
>   cat("Erreur L2 =",sum((RPL)^2)
Erreur L2 = 5.476764
```

[5] revient longuement sur l'analyse des résidus dans le cadre de modèles linéaires généralisés.

Rappelons que l'outil de base pour quantifier la qualité de la régression est la *déviance*

$$D(\boldsymbol{\beta}) = -2[\log \mathcal{L}(\widehat{\boldsymbol{\beta}}|Y) - \log \mathcal{L}_\star(Y)]$$

où $\log \mathcal{L}(\boldsymbol{\beta}|Y)$ désigne la log-vraisemblance du modèle, et où $\log \mathcal{L}_\star(Y)$ est la log-vraisemblance saturée (obtenue avec un modèle parfait).

```
> logLik(regPlog)
'log Lik.' -7.955383 (df=2)
> deviance(regPlog)
[1] 1.760214
> AIC(regPlog)
[1] 19.91077
> -2*logLik(regPlog)+2*2
[1] 19.91077
attr(,"df")
```

Dans un souci de partimonie, on pénalise souvent log-vraisemblance par le nombre de paramètres, ce qui correspond au critère d'information d'Akaike (AIC, en multipliant par 2). On peut également définir le critère de Schwartz,

$$\begin{cases} AIC : -2\log \mathcal{L}(\widehat{\boldsymbol{\beta}}) + 2k \\ BIC : -2\log \mathcal{L}(\widehat{\boldsymbol{\beta}}) + k\log(n) \end{cases}$$

Il existe aussi un critère d'Aikaike *corrigé* (introduit par [15]) dans le cas où l'on a trop peu d'observations. Toutes ces fonctions peuvent être obtenues à l'aide de la fonction `AIC` de `library(aod)` ou `BIC` de `library(BMA)`, ou encore `extractAIC` avec comme paramètre `k=log(nrow(base))`.

```
>   cat("AIC (Poisson-log) =",extractAIC(regPlog,k=2)[2])
AIC (Poisson-log) =  19.91077
>   cat("BIC (Poisson-log) =",extractAIC(regPlog,k=log(nrow(base)))[2])
BIC (Poisson-log) =  19.12964
```

On peut comparer tous les modèles via :

```
> AIC(regNId,regNlog,regPId,regPlog,regGId,regGlog)
        df      AIC
regNId   3 21.10099
regNlog  3 20.63884
regPId   2 19.86546
regPlog  2 19.91077
regGId   3 18.01344
regGlog  3 18.86736
```

1.1.7 Les variables tarifaires continues et la nonlinéarité

Le but de la tarification (et plus généralement de toute prédiction) est d'estimer une espérance conditionnelle,

$$\mathbb{E}(S|\boldsymbol{X}=\boldsymbol{x})=\varphi(\boldsymbol{x}) \text{ ou } S=\varphi(X_1,\cdots,X_k)+\varepsilon$$

où $\varphi : R^k \to R$. Supposer un modèle *linéaire* est problement une hypothèse trop forte. Mais on se doute qu'estimer une fonction définie sur $\mathbb{R}^k$ serait trop complexe numériquement. Un bon compromis est proposé par les modèles dit *additifs*.

A titre d'illustration, la Figure 1.3 permet de visualiser l'impact de la densité de population dans la commune de l'assuré sur la fréquence de sinistre. Les points noirs correspondent à la fréquence moyenne empirique observée pour différents niveaux de densité

```
> library(mgcv)
> reg.gam <- gam(nombre~s(densite),offset=log(exposition),
+ family=poisson(link="log"),data=sinistres)
> dens.x <- seq(0,30000,100)
> pred <- predict(reg.gam,newdata=data.frame(densite=dens.x,expo=1),
+ se=TRUE,type="response")
> plot(dens,pred$fit,col="blue",lwd=2)
> lines(dens,pred$fit+2*N1RC0as1$se.fit,col="red",lty=2)
> lines(dens,pred$fit-2*N1RC0as1$se.fit,col="red",lty=2)
```

Les modèles GAM

Les modèles additifs ont été introduits par [30] qui notait qu'estimer une fonction $\varphi : \mathbb{R}^k \to \mathbb{R}$ serait numériquement trop complexe (et probablement peu robuste). On cherche ici une décomposition de la forme

$$S=\varphi_1(X_1)+\cdots+\varphi_k(X_k)+\varepsilon$$

où les fonctions $\varphi_j : \mathbb{R} \to \mathbb{R}$ sont supposées suffisament régulières. En fait, ce modèle n'est valable que pour les variables X_j continues, les variables qualitatives continuant - généralement - à intervenir sous une forme linéaire. Autrement dit, un modèle additif serait

$$S=\varphi_1(X_1)+\beta_2 X_2+\varepsilon$$

où X_1 est l'âge du conducteur, et X_2 le carburant du véhicule. Notons qu'il serait aussi possible de considérer un modèle de la forme

$$S=\begin{cases}\varphi_{1,E}(X_1)+\varepsilon & \text{si } X_2=\text{ essence}\\ \varphi_{1,D}(X_1)+\varepsilon & \text{si } X_2=\text{ diesel}\end{cases}$$

Ces deux types de modèles sont estimés ci-dessous.

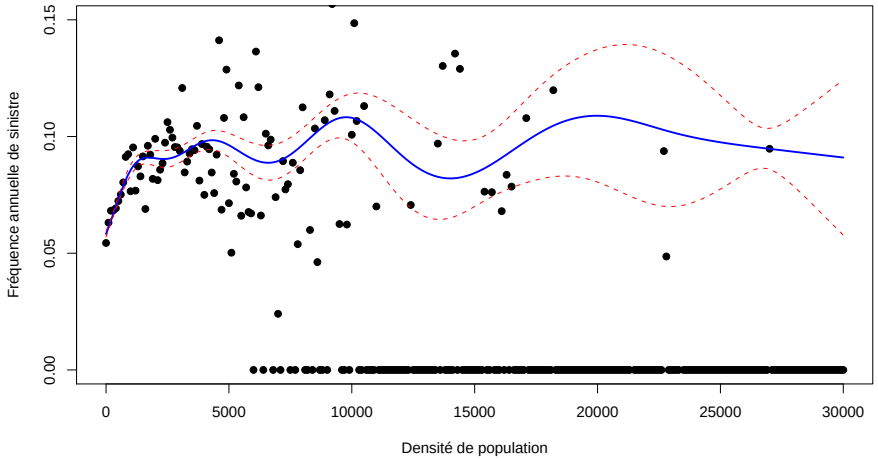


FIGURE 1.3 – Fréquence individuelle en fonction de la densité de population de la commune de résidence du conducteur principal.

```
> library(mgcv)
> reg <- gam(nombre~s(ageconducteur)+offset(exposition),
+ data=sinistres,family=poisson)
> age <- seq(17,100)
> AGE <- data.frame(ageconducteur=age,exposition=1)
> Y <- predict(reg,AGE,type="response")
> reg = gam(nombre~s(ageconducteur)+carburant+offset(exposition),
+ data=sinistres,family=poisson)
> AGE <- data.frame(ageconducteur=age,exposition=1,carburant="E")
> YE <- predict(reg,AGE,type="response")
> AGE <- data.frame(ageconducteur=age,exposition=1,carburant="D")
> YD <- predict(reg,AGE,type="response")
> plot(age,Y,type='l')
> lines(age,YD,col='blue')
> lines(age,YE,col='red')
```

Pour le premier type de modèle, ou le code suivant pour le second,

```
> library(mgcv)
> reg <- gam(nombre~s(ageconducteur)+offset(exposition),
+ data=sinistres,family=poisson)
> age <- seq(17,100)
> AGE <- data.frame(ageconducteur=age,exposition=1)
> Y <- predict(reg,AGE,type="response")
```

```
> reg <- gam(nombre~s(ageconducteur)+offset(exposition),
+ data=sinistres[sinistres$carburant=="E",],family=poisson)
> YE <- predict(reg,AGE,type="response")
> reg <- gam(nombre~s(ageconducteur)+offset(exposition),
+ data=sinistres[sinistres$carburant=="D",],family=poisson)
> YD=predict(reg,AGE,type="response")
> plot(age,Y,type='l')
> lines(age,YD,col='blue')
> lines(age,YE,col='red')
```

Ce petit exemple montre bien les limites de ces modèles additifs.

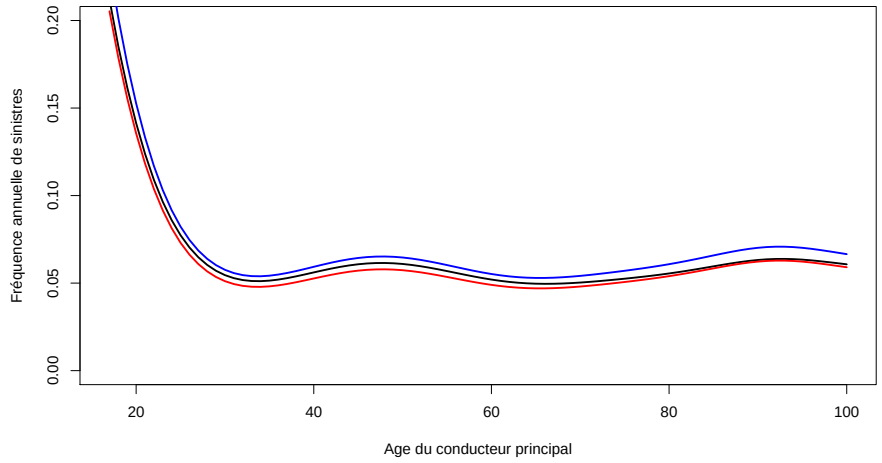


FIGURE 1.4 – Modèle GAM addifif, $S = \varphi_1(X_1) + \beta_2 X_2 + \varepsilon$ où X_2 désigne le type de carburant.

L'estimation de ces modèles peut se faire de plusieurs manières sous R. Il y a tout d'abord la fonction `gam` de `library(gam)`, basé sur l'algorithme proposé par [13]. La fonction `gam` de `library(mgcv)` repose sur la méthodologie développée par [32]. Enfin d'autres packages proposent aussi des estimations de ces transformations nonlinéaires, dont `library(gmlss)` ou `library(gss)`.

Une autre possibilité est également d'uiliser la fonction `glm` avec la `library(splines)`. On peut alors changer facilement le nombre de degrés de liberté, i.e. le paramètre de lissage de la transformation,

```
> library(splines)
> reg3 <- glm(nombre~bs(ageconducteur,df=3)+offset(exposition),
+ data=nombre,family=poisson)
```

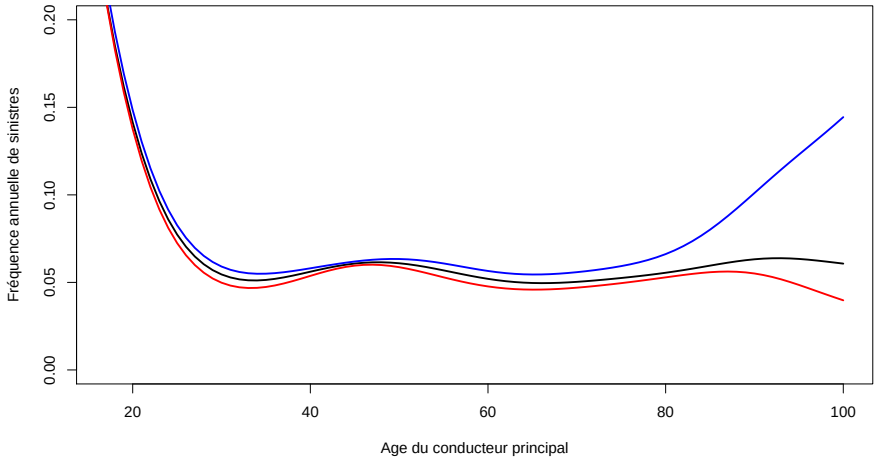


FIGURE 1.5 – Modèle GAM, $S = \begin{cases} \varphi_{1,E}(X_1) + \varepsilon & \text{si } X_2 = \text{ essence} \\ \varphi_{1,D}(X_1) + \varepsilon & \text{si } X_2 = \text{ diesel} \end{cases}$ où X_2 désigne le type de carburant.

La Figure 1.6 montre ainsi la prédiction de la fréquence moyenne en fonction de l'âge, avec différents paramètres de lissage.

Les modèles MARS

Une autre classe de modèle particulièrement intéressant a été présentée par [10], appelés MARS, *Multiplicative Adaptative Regression Splines*. On considère ici une base de fonctions de φ de la forme $(\pm(x-k)_+)$.

En particulier, par rapport à un modèle linéaire simple $Y = \beta_0 + \beta_1 X + \varepsilon$, on considère ici un modèle avec rupture,

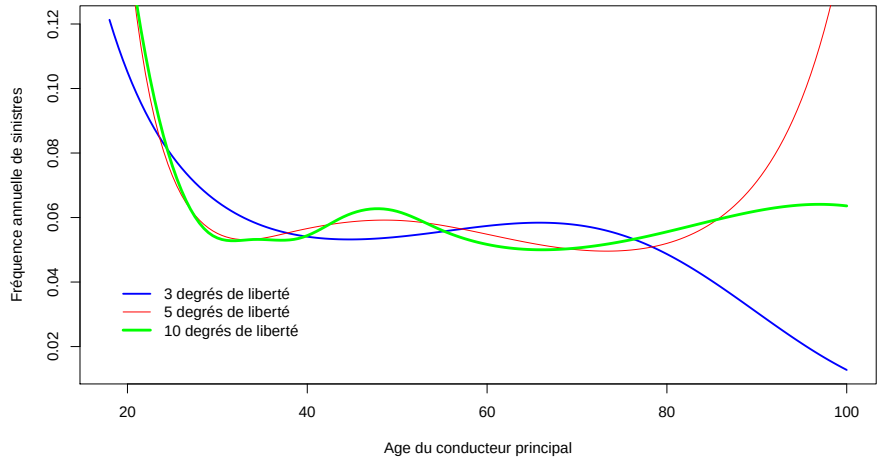
$$Y = \beta_0 + \beta_1 \max\{0, X-k\} + \beta_1 \max\{0, k-X\} + \varepsilon$$

où k devient également un paramètre à estimer.

```
> library(mda)
> reg <- mars(sinistres$ageconducteur,sinistres$nombre,nk=10)
> summary(lm(sinistres$nombre~reg$x-1))

Call:
lm(formula = sinistres$nombre ~ reg$x - 1)

Residuals:
     Min       1Q   Median       3Q      Max
```




```
-0.08342 -0.03916 -0.03730 -0.03560 15.96203

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
reg$x1 3.408e-02  4.528e-04  75.271   <2e-16 ***
reg$x2 1.692e-04  2.007e-05   8.432   <2e-16 ***
reg$x3 4.486e-03  1.694e-04  26.477   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.2071 on 678010 degrees of freedom
Multiple R-squared: 0.03526,    Adjusted R-squared: 0.03526
F-statistic:  8261 on 3 and 678010 DF,  p-value: < 2.2e-16
> age <- seq(17,100)
> Y <- predict(reg,age)
> plot(age,Y)
```

1.1.8 Les modèles nonlinéaires multivariés

On peut s'autoriser éventuellement encore un peu plus de souplesse en prenant en compte le couple constitué de deux variables continues,

$$S = \varphi(X_1, X_2) + \varepsilon$$

où $\varphi : \mathbb{R}^2 \to \mathbb{R}$, au lieu d'un modèle GAM classique,

$$S = \varphi_1(X_1) + \varphi_2(X_2) + \varepsilon$$

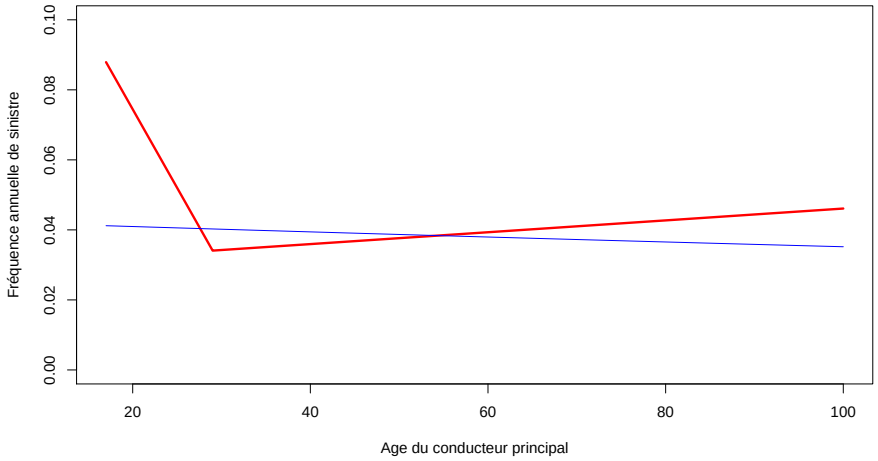


FIGURE 1.7 – Modèle MARS, impact de l'âge du conducteur principal sur la fréquence de sinistres.

Cette option est proposée par exemple dans `library(mgcv)`

1.2 Modéliser des variables indicatrices

Les bases des modèles GLM étant posées, nous allons les utiliser en tarification, en modélisant tout d'abord des variables indicatrices 0/1 dans un premier temps, avant de modéliser la fréquence de sinistres, puis les coûts individuels dans les prochaines sections.

Remarque 1.2 *Les modèles sont très utilisés en techniques de scoring afin de savoir s'il convient d'occtroyer un crédit à quelqu'un.*

1.2.1 La régression logistique ou probit

La régression logistique suppose que si $\pi(Y|\boldsymbol{X}) = \mathbb{P}(Y = 1|\boldsymbol{X})$, alors

$$\frac{\pi(Y|\boldsymbol{X})}{1-\pi(Y|\boldsymbol{X})} = \frac{\mathbb{P}(Y=1|\boldsymbol{X})}{\mathbb{P}(Y=0|\boldsymbol{X})} = \exp(\boldsymbol{X}\boldsymbol{\beta})$$

Dans le cas du modèle probit, on suppose qu'il existe un modèle latent Gaussien, tel que

$$Y_i^\star = \boldsymbol{X}_i'\boldsymbol{\beta} + \varepsilon_i$$

et que $Y_i = 0$ si $Y_i^\star < s$, et $Y_i = 1$ si $Y_i^\star > s$, et $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

La synthaxe de ces deux modèles est très proche, car seule la fonction de lien change.

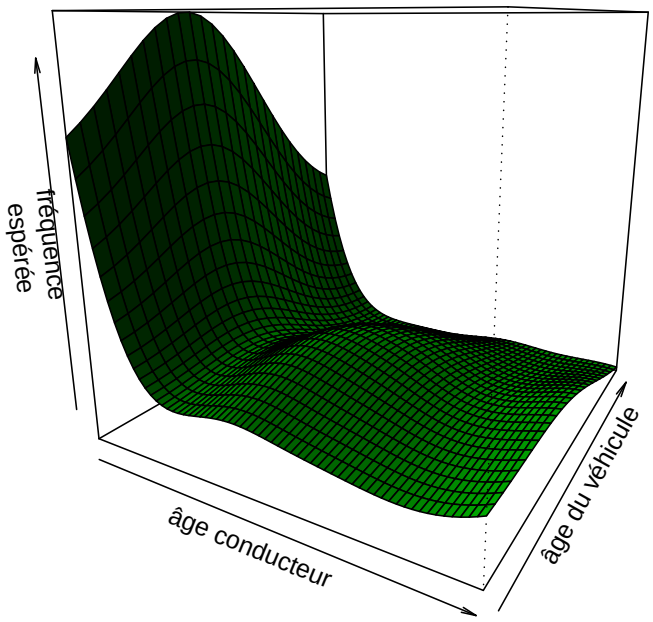


FIGURE 1.8 – Fréquence prédite $\widehat{Y}$, en fonction de l'âge du conducteur et de l'ancienneté du véhicule, $\widehat{Y} = \widehat{\varphi}(X_1, X_2)$.

```
> sinistres$touche <- sinistres$nombre>0
> reglogit <- glm(touche~ageconducteur,
+ data=sinistres,family=binomial(link="logit"))
> regprobit <- glm(touche~ageconducteur,
+ data=sinistres,family=binomial(link="probit"))
> age <- seq(17,100)
> AGE <- data.frame(ageconducteur=age,exposition=1)
> Yl <- predict(reglogit,AGE,type="response")
> Yp <- predict(regprobit,AGE,type="response")
> plot(age,Yp-Yl,type="l")
> abline(h=0,lty=2)
```

On notera que ces deux modèles donnent des prédictions très proches, comme le montre la Figure 1.13.

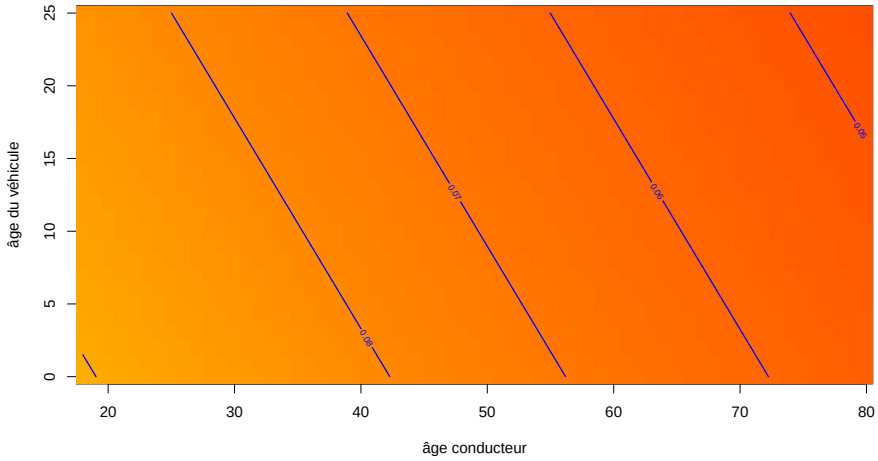


FIGURE 1.9 – Fréquence prédite $\widehat{Y}$ par un modèle GLM $\widehat{Y} = \exp(\widehat{\beta}_0 + \widehat{\beta}_1 X_1 + \widehat{\beta}_2 X_2)$.

1.2.2 Les arbres de régression

Les arbres de régression sont des outils nonparamétriques de segmentation. Dans un arbre de décision, on cherche à détecter des critères permettant de répartir les individus en 2 classes, caractérisées par $Y = 0$ et $Y = 1$. On commence par choisir la variable, qui, par ses modalités, sépare le mieux les individus de chacune des classes. On constitue alors un premier *noeud*. On réintère alors la procédure sur chaque nouveau noeud. Dans la méthode CART (*Classification And Regression Tree*), on regarde toutes les possibilités. On continue soit jusqu'à ce qu'il ne reste plus qu'un seul. individu dans chaque noeud, soit suivant un critère d'arrêt. Les critères de discrimination et de constitution des noeuds sont généralement les suivants,

- lorsque les variables explicatives X_j sont qualitatives, ou discrètes, on utilise la distance du χ^2 (on parle d'arbre CHAID),
- en présence de variables de tous types, on peut utiliser l'indice de Gini (méthode CART),
- ou l'entropie (méthode C5.0),

Pour un varible continue, on distinguera $\{X_1 \leq s\}$ et $\{X_1 > s\}$. Pour une variable qualitative, on distinguera $\{X_1 = x\}$ et $\{X_1 \neq x\}$.

Pour chacune des variables, on regarde l'ensemble des classifications possibles. Quelles que soient les variables, on définit :

```
> seuilagecond <- unique(nombre$ageconducteur)
> seuilregion  <- unique(nombre$region)
```

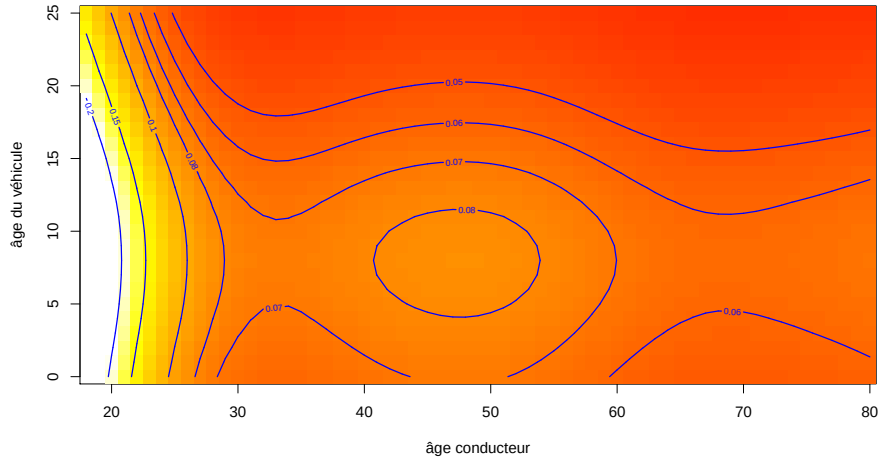


FIGURE 1.10 – Fréquence prédite $\widehat{Y}$ par un modèle additif $\widehat{Y} = \widehat{\varphi}_1(X_1) + \widehat{\varphi}_2(X_2)$.

Pour les variables quantitatives, on distingue :

```
> k=5
> classe0 <- nombre$ageconducteur<=seuilagecod[k]
> classe1 <- nombre$ageconducteur>seuilagecod[k]
```

alors que pour les variables qualitatives,

```
> k=5
> classe0 <- nombre$region==seuilregion[k]
> classe1 <- nombre$region!=seuilregion[k]
```

Une fois constituées les 2 classes, on calcule un des critères possibles.

Si on regarde la décomposition obtenue sur le premier noeud, on observe que pour les conducteurs de moins de 25 ans, la probabilité d'avoir un accident est de 10%, contre 5% pour les conducteurs de plus de 25 ans. Dans le cas des régions, avec une distance du chi-deux, on cherche à minimiser

$$\chi^2 = - \sum_{\text{classe}\in\{0,1\}} \sum_{y\in\{0,1\}} \frac{[n_{\text{classe},y} - n^{\perp}_{\text{classe},y}]^2}{n^{\perp}_{\text{classe},y}}$$

où $n_{\text{classe},y}$ désigne le nombre de personnes dans la classe considérée pour lesquelles la variable Y prend la modalité y.

```
> base=sinistres[sinistres$ageconducteur<=85,]
> seuil=sort(unique(base$ageconducteur))
```

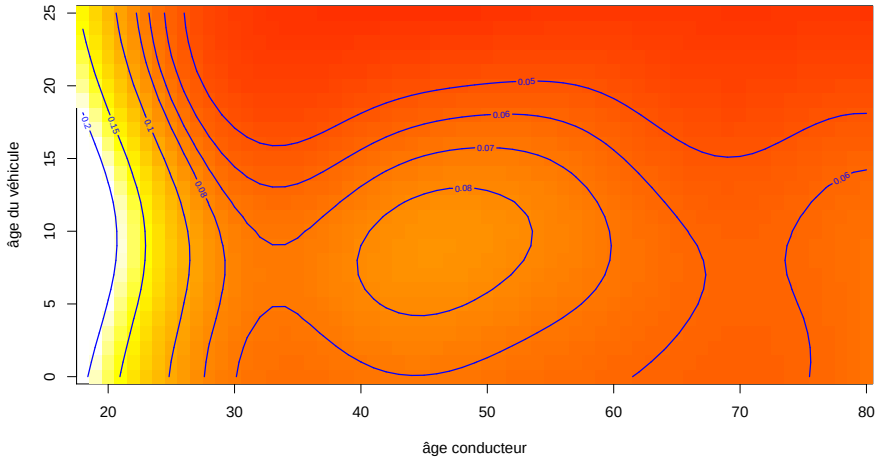


FIGURE 1.11 – Fréquence prédite $\widehat{Y}$ par un modèle additif $\widehat{Y} = \widehat{\varphi}(X_1, X_2)$.

```
> TABLE=rep(NA,length(seuil))
> names(TABLE)=seuil
> for(k in 1:(length(seuil)-1)){
+ classe0 <- base$ageconducteur<=seuil[k]
+ classe1 <- base$ageconducteur>seuil[k]
+ M=matrix(
+ rbind(c(sum(base$touche[classe0]==FALSE),
+          sum(base$touche[classe0]==TRUE)),
+        c(sum(base$touche[classe1]==FALSE),
+          sum(base$touche[classe1]==TRUE))),2,2)
+ TABLE[k]=-chisq.test(M)$statistic
}
> which.min(TABLE)
23
 6
> plot(seuil,TABLE)
```

Autrement dit le meilleur découpage possible est `(17,23]` et `(23,85]`

A la seconde étape, on cherche une autre partition, en considérant la précédente comme acquise,

```
> k1 = which(seuil==23)
> for(k in 1:(length(seuil)-1)){
+ if(k!=k1){
+ classe0 <- (base$ageconducteur<=seuil[k])&(base$ageconducteur<=seuil[k1])
```

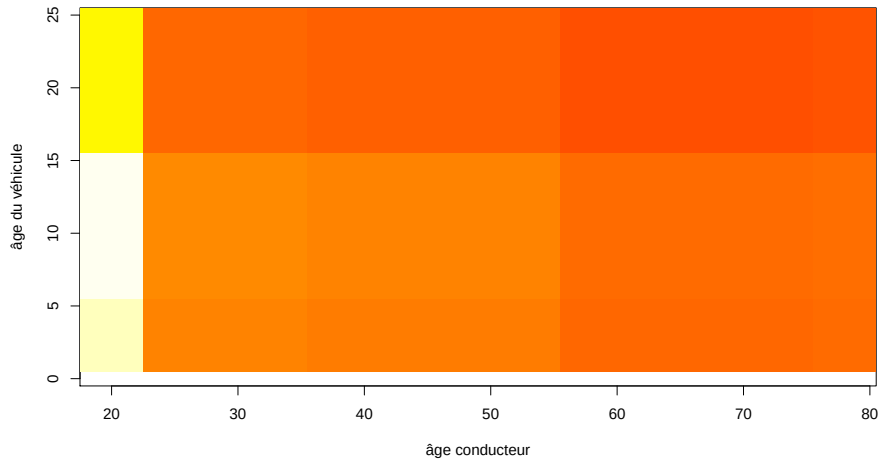


FIGURE 1.12 – Fréquence prédite $\widehat{Y}$ par un modèle par classes jointes, $(X_1, X_2) \in [a_1, b_1] \times [a_2, b_2]$.

```
+ classe2 <- (base$ageconducteur>seuil[k])&(base$ageconducteur>seuil[k1])
+ classe1 <- 1-classe0-classe2
+ M=matrix(
+ rbind(c(sum(base$touche[classe0]==FALSE),
+          sum(base$touche[classe0]==TRUE)),
+        c(sum(base$touche[classe1]==FALSE),
+          sum(base$touche[classe1]==TRUE)),
+        c(sum(base$touche[classe2]==FALSE),
+          sum(base$touche[classe2]==TRUE))),3,2)
+ TABLE[k]=-chisq.test(M)$statistic
+ }}
> which.min(TABLE)
84
67
> plot(seuil,TABLE)
```

En l'occurence, on ne nous conseille ici pas d'autre classe (ou alors à un âge très avancé). On retrouvera ce découpage en deux classes dans la section sur les modèles MARS par exemple.

Parmi les autres critères, on peut aussi utiliser la distance de Gini,

$$G = -\sum_{\text{classe}\in\{0,1\}} \frac{n_{\text{classe}}}{n} \sum_{y\in\{0,1\}} \frac{n_{\text{classe},y}}{n_{\text{classe}}} \left(1 - \frac{n_{\text{classe},y}}{n_{\text{classe}}}\right)$$

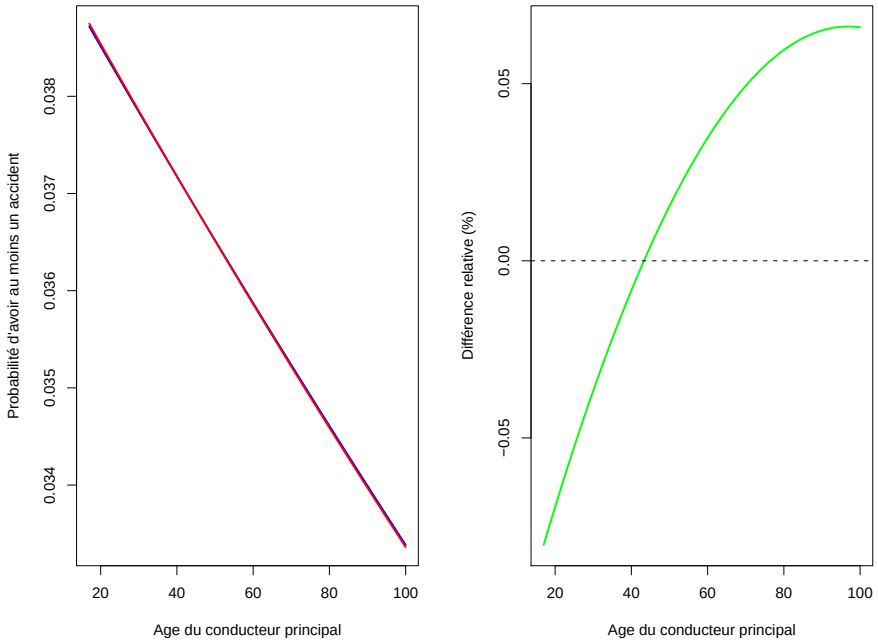


FIGURE 1.13 – Régression logistique (logit) versus modèle latent Gaussien (probit) pour prédire la probabilité d'avoir au moins un accident dans l'année, en fonction de l'âge du conducteur principal.

ou l'entropie,

$$E = -\sum_{\text{classe}\in\{0,1\}} \frac{n_{\text{classe}}}{n} \sum_{y\in\{0,1\}} \frac{n_{\text{classe},y}}{n_{\text{classe}}} \log\left(\frac{n_{\text{classe},y}}{n_{\text{classe}}}\right)$$

Les arbres permettent une lecture relativement aisée pour l'utilisateur, et reposent sur des techniques nonparamétriques. Aussi, contrairement aux méthodes GLM que nous verrons par la suite, le choix des lois ou la recherche d'éventuelles nonlinéarités n'intervient pas ici. Les arbres sont également peu sensibles aux *outliers*. Mais les arbres, de par leur construction, posent aussi certains soucis. En particulier, on ne peut pas revenir en arrière, et le séquencement est très important.

1.2.3 Probabilité d'avoir (au moins) un sinistre dans l'année

A titre d'illustration, étudions la probabilité d'avoir au moins un sinistre dans l'année. Par défaut, l'arbre ne permet pas de définir des classes, et on obtient autant de classes que l'on a d'âges,

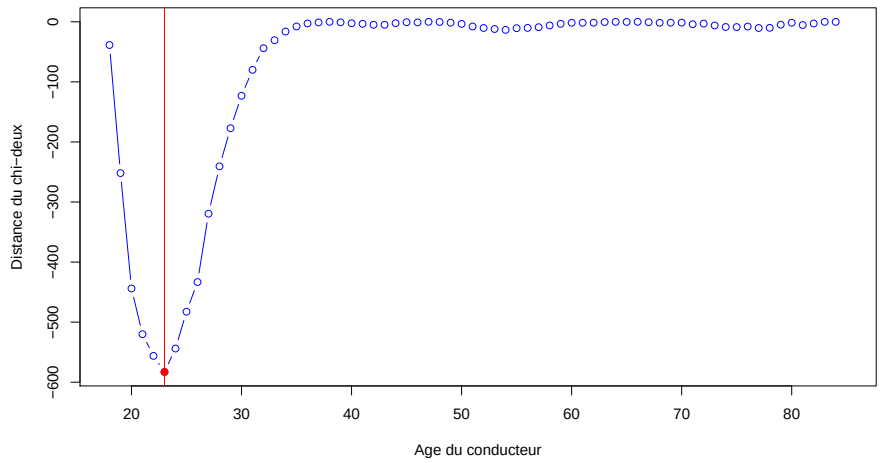


FIGURE 1.14 – Evolution de χ^2 lors du découpage en 2 classes `(17,k]` et `(k,85]`.

```
> library(tree)
> arbre=tree((nombre>0)~ageconducteur,data=sinistres,split="gini")
> age=data.frame(ageconducteur=18:90)
> y=predict(arbre,age)
> plot(age$ageconducteur,y)
```

Si l'on souhaite coupe les branches de l'arbre, on peut utiliser l'option `mincut` pour dire qu'on ne peut couper davantage qu'à condition de constituer des classes dont le nombre d'invidus à l'intérieur soit suffisement élevé.

```
> arbre2=tree((nombre>0)~ageconducteur,data=sinistres,split="gini",
+ mincut = 20000)
> y2=predict(arbre2,age)
> lines(age$ageconducteur,y2,col="red",type="s",lwd=2)
> arbre3=tree((nombre>0)~ageconducteur,data=sinistres,split="gini",
+ mincut = 100000)
> y3=predict(arbre3,age)
> lines(age$ageconducteur,y3,col="purple",type="s",lwd=2)
```

On obtient alors les classes décrites sur la figure 1.16.

1.2.4 Probabilité d'avoir un gros sinistre dans l'année

Cette étude sera particulièrement intéressante pour écrêter les gros sinistres (nous reviendrons sur ce point dans la section 1.4.3). On supposera

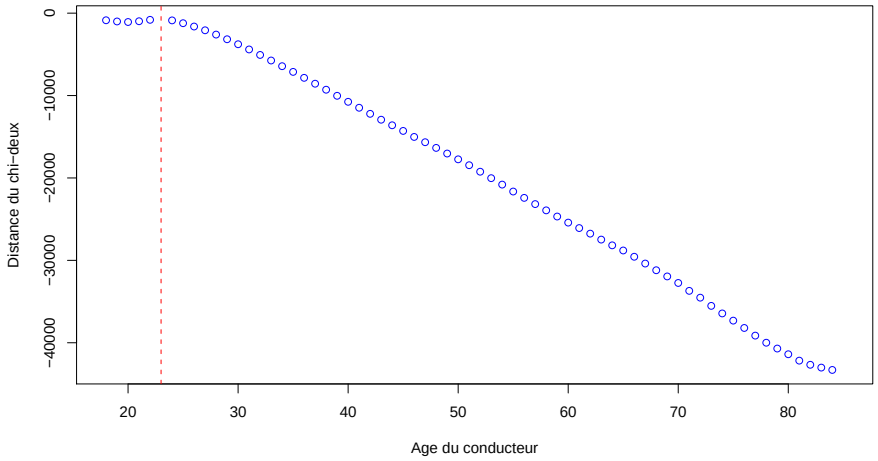


FIGURE 1.15 – Evolution de χ^2 lors du découpage en 3 classes, `(17,23]`, `(23,k]` et `(k,85]`, ou `(17,k]`, `(k,23]` et `(23,85]`.

(arbitrairement) que les *gros* sinistres sont ceux dont le montant dépasse 50 000 euros.

```
> library(tree)
> sinistres$GS <- sinistres$cout>50000
> ARBRE <- tree(GS ~ puissance + zone + agevehicule ,
+ data=sinistres,split="gini")
```

Cet arbre étant manifestement trop étendu, on peut limiter en demandant à avoir au moins 5 000 assuré par branche,

```
> ARBRE <- tree(GS ~ puissance + zone + agevehicule ,
+ data=sinistres,split="gini",minsize = 5000)
> ARBRE
> ARBRE
node), split, n, deviance, yval
      * denotes terminal node

 1) root 26444 87.710 0.003328
   2) zone: B,C,D,E,F 23080 68.790 0.002990
     4) puissance < 5.5 8028 17.960 0.002242
       8) zone: B,D,F 3442  3.995 0.001162 *
       9) zone: C,E 4586 13.960 0.003053 *
     5) puissance > 5.5 15052 50.830 0.003388
```

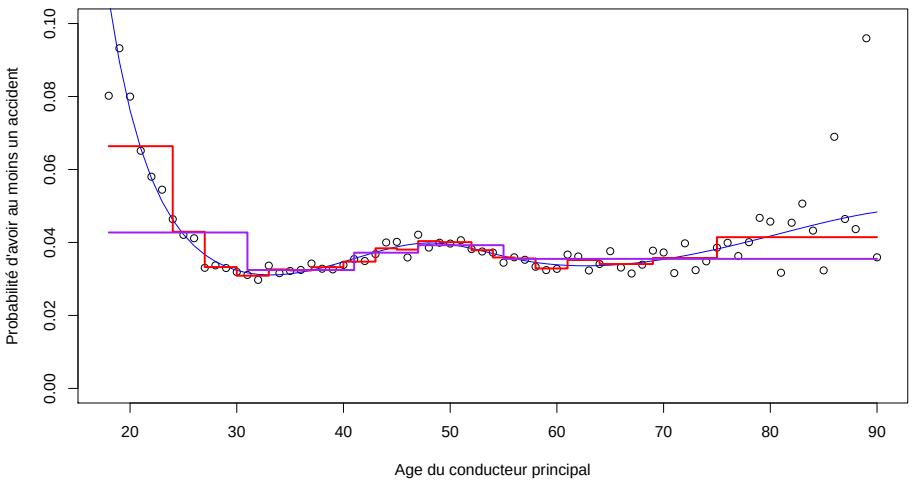


FIGURE 1.16 – Prédiction par arbre de régression, avec plus ou moins de classes d'âge.

```
    10) zone: B,C,E 10372 30.910 0.002989
      20) agevehicule < 10.5 7541 17.960 0.002387
        40) puissance < 7.5 5274 14.960 0.002844
          80) agevehicule < 2.5 1291  5.972 0.004648 *
          81) agevehicule > 2.5 3983  8.980 0.002260 *
        41) puissance > 7.5 2267  2.996 0.001323 *
      21) agevehicule > 10.5 2831 12.940 0.004592 *
    11) zone: D,F 4680 19.910 0.004274 *
  3) zone: A 3364 18.890 0.005648 *
```

On note qu'en fonction de la zone, de la puissance du véhicule et de l'ancienneté du véhicule, on peut déterminer avec une bonne assurance la probabilité d'avoir un très gros sinistre. Par exemple, pour les personnes n'habitant pas un endroit trop dense (les zones les plus denses correspondant à `zone=A`), en particulier les zones `B`, `D` et `E`, et si la puissance n'est pas trop élevée, `puissance<5.5` la probabilité d'avoir un très gros sinistres est de l'ordre de 1/1000. La probabilité sera 4 fois plus grande si la le véhicule est puissant (`puissance>5.5`) et ancien, (`agevehicule>10.5`). Dans une zone dense, la probabilité sera plus de 5 fois plus grande (quelles que soient les autres variables).

Si on trace l'arbre, on obtient le dessin de la Figure 1.17

```
> plot(ARBRE)
> text(ARBRE,cex=.9,col="blue")
```

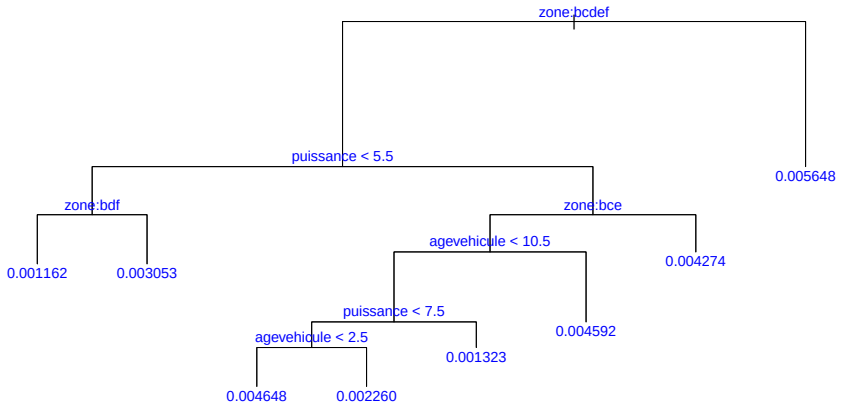


FIGURE 1.17 – Arbre de prédiction, pour expliquer la probabilité d'avoir (ou pas) un gros sinistre, en fonction de la densité de population, de l'ancienneté du véhicule, et de sa puissance.

1.3 Modéliser la fréquence de sinistralité

Dans cette section, nous allons rentrer davantage dans la modélisation par modèles linéaires généralisés. Mais avant de commencer, il peut être intéressant de regarder un peu la base, et de faire un peu d'analyse descriptive pour comprendre la loi du nombre de sinistres par contrat.

1.3.1 Un peu d'analyse descriptive

La fréquence de sinistres

Une hypothèse forte de la loi de Poisson est que $\mathbb{E}(N) = \text{Var}(N)$

Si l'on compare les valeurs numériques, cela donne l'ajustement suivant, si l'on estime le paramètre par la méthode des moments (ou par maximum de vraisemblance, `ML` qui ici coïncident) :

```
> library(vcd)
> gof = goodfit(N,type= "poisson",method= "ML")
> gof

Observed and fitted values for poisson distribution
with parameters estimated by 'ML'
```

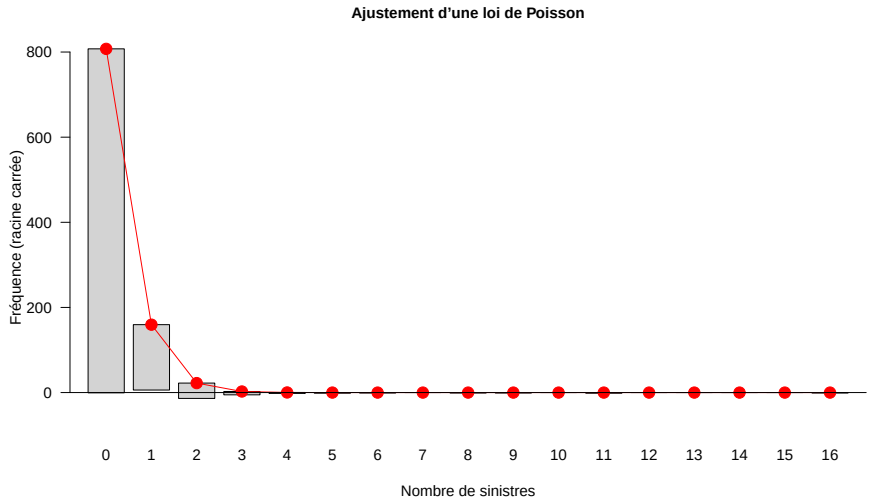


FIGURE 1.18 – Fréquence empirique du nombre de sinistres par police d'assurance.

	count	observed	fitted
[1,]	653047	653047	652055
[2,]	23592	23592	25453
[3,]	1299	1299	496
[4,]	62	62	6
[5,]	5	5	0
[6,]	2	2	0
[7,]	1	1	0
[8,]	0	0	0
[9,]	1	1	0
[10,]	1	1	0
[11,]	0	0	0
[12,]	2	2	0
[13,]	0	0	0
[14,]	0	0	0
[15,]	0	0	0
[16,]	0	0	0
[17,]	1	1	0

La différence entre la valeur prédite par le modèle Poissonnien et les valeurs observées nous poussent à essayer de mieux comprendre l'hétérogénéité qui existe au sein de nos données.

Les variables qualitatives, ou facteurs

Les facteurs sont des codifications de variables *qualitatives*. Dans la base, nous disposons de plusieurs variables qualitatives comme le carburant `carburant` codé en `E` pour essence et `D` pour diesel, ou encore `region` pour la région française (visualisées sur la Figure 1.19)

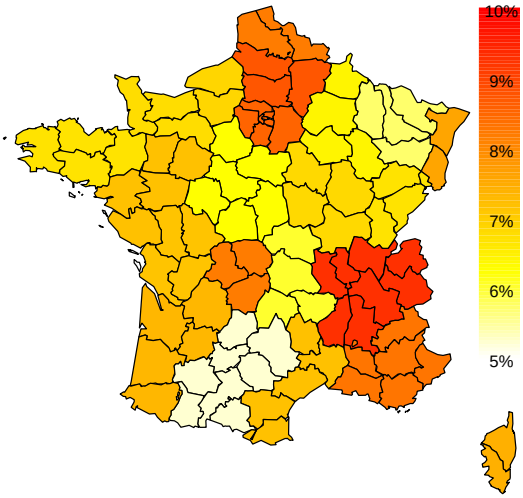


FIGURE 1.19 – Fréquence empirique observée par région française.

Régresser une variable quantitative (comme le nombre de sinistres) sur une variable factorielle correspond à faire une *analyse de la variance*.

```
> summary(lm(nombre~as.factor(region), data=nombre))

Call:
lm(formula = nombre ~ as.factor(region), data = nombre)

Residuals:
     Min       1Q   Median       3Q      Max
-0.16115 -0.03477 -0.03477 -0.03477 15.96523

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)          0.161150   0.002341   68.84   <2e-16 ***
as.factor(region)0  -0.091614   0.002763  -33.16   <2e-16 ***
as.factor(region)1  -0.102853   0.002819  -36.48   <2e-16 ***
as.factor(region)2  -0.113822   0.002815  -40.43   <2e-16 ***
as.factor(region)3  -0.112168   0.002796  -40.12   <2e-16 ***
as.factor(region)4  -0.112059   0.002760  -40.60   <2e-16 ***
as.factor(region)5  -0.115453   0.002737  -42.18   <2e-16 ***
as.factor(region)6  -0.116853   0.002711  -43.10   <2e-16 ***
as.factor(region)7  -0.119643   0.002688  -44.51   <2e-16 ***
as.factor(region)8  -0.131576   0.002798  -47.03   <2e-16 ***
as.factor(region)9  -0.129934   0.002818  -46.11   <2e-16 ***
as.factor(region)10 -0.133945   0.002804  -47.77   <2e-16 ***
as.factor(region)11 -0.134594   0.002818  -47.77   <2e-16 ***
as.factor(region)12 -0.134683   0.002858  -47.12   <2e-16 ***
as.factor(region)13 -0.126384   0.002362  -53.50   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.2067 on 677998 degrees of freedom
Multiple R-squared: 0.005699,   Adjusted R-squared: 0.005678
F-statistic: 277.6 on 14 and 677998 DF,  p-value: < 2.2e-16
```

ou directement, à l'aide de la fonction `aov`,

```
> summary(aov(nombre~as.factor(region), data=nombre))
                     Df Sum Sq Mean Sq F value    Pr(>F)
as.factor(region)     14    166 11.8542  277.56 < 2.2e-16 ***
Residuals         677998  28956  0.0427
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

qui montre que la fréquence de sinistres est sensiblement différente d'une région à l'autre.

Les variables quantitatives continues

Parmi les variables continues permettant d'expliquer la fréquence de sinistres, on retiendra l'âge du véhicule `agevehicule`, ou l'âge du conducteur `ageconducteur`. On pourrait également utiliser le bonus observé à la souscription du contrat `bonus`. Cette variable est liée à l'ancienneté du permis et peut s'avérer délicate à prendre en compte dans le modèle.

La Figure 1.20 montre la fréquence empirique obervée en fonction de l'âge du conducteur principal (fréquence brute).

```
> age = seq(18,100,by=1)
> FREQ = rep(NA,length(age))
```

```
> for(k in 1:length(FREQ)){
+ I=nombre$ageconducteur==age[k]
+ X=nombre$nombre[I]
+ W=nombre$exposition[I]
+ FREQ[k]=weighted.mean(X/W,W)
+ }
> plot(age,FREQ)
```

La moyenne empirique est ici corrigée par l'exposition. La fréquence annuelle devrait être le nombre de sinistres observé divisé par l'exposition, et on met un poids proportionnel à l'exposition.

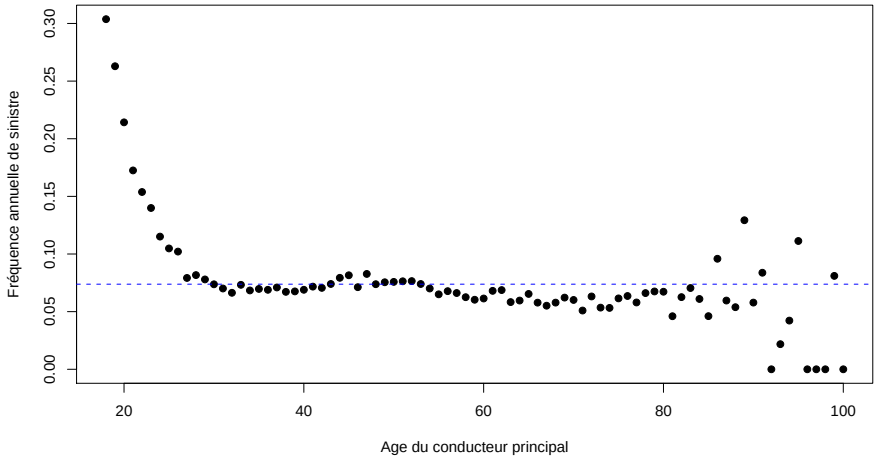


FIGURE 1.20 – Fréquence empirique par âge du conducteur principal. Le trait horizontal montre la fréquence moyenne d'un individu pris au hasard.

1.3.2 La méthode des marges

[1] a proposé une méthode relativement simple pour faire de la tarification, appelée *method of marginal totals*. Avant de présenter cette méthode, notons que [16] a retrouvé cette méthode en faisant du maximum de vraisemblance sur un modèle Poissonnien. Plaçons nous dans le cas où les variables exogène $\boldsymbol{X}$ sont qualitatifs, de telle sorte que l'on puisse définir des *classes de risques*. Alors

$$\mathbb{P}(N = n|\boldsymbol{X} = \boldsymbol{X}) = \exp[-\lambda_{\boldsymbol{X}}]\frac{\lambda_{\boldsymbol{X}}^n}{n!} \text{ où } \lambda_{\boldsymbol{X}} = \exp[-\boldsymbol{X}'\boldsymbol{\beta}]$$

ce qui donne une log-vraisemblance de la forme

$$\mathcal{L}(\boldsymbol{\beta}|n_i, \boldsymbol{X}_i) = \sum_{i=1}^{n} [-\lambda_{\boldsymbol{X}_i}] + n_i \log[\lambda_{\boldsymbol{X}_i}] - \log[n_i!]$$

dont la condition du premier ordre donne les équations normales,

$$\sum_{i,\boldsymbol{X}_i=\boldsymbol{X}} n_i = \sum_{i,\boldsymbol{X}_i=\boldsymbol{X}} \lambda_{\boldsymbol{X}}$$

pour toute classe de risque $\boldsymbol{X}$.

Si on regarde le cas où les classes de risque sont constitués par la puissance du véhicule (définie en tant que facteur),

```
> nombre$puissance=as.factor(nombre$puissance)
> marges=glm(nombre~puissance,
+ data=nombre,family=poisson(link="log"))
> summary(marges)

Call:
glm(formula = nombre ~ puissance, family = poisson(link = "log"),
    data = nombre)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
puissance4  -3.35967    0.01580 -212.70   <2e-16 ***
puissance5  -3.19353    0.01397 -228.53   <2e-16 ***
puissance6  -3.16181    0.01259 -251.14   <2e-16 ***
puissance7  -3.25744    0.01337 -243.68   <2e-16 ***
puissance8  -3.34965    0.02463 -135.98   <2e-16 ***
puissance9  -3.20436    0.02862 -111.97   <2e-16 ***
puissance10 -3.24813    0.02865 -113.36   <2e-16 ***
puissance11 -3.24661    0.03742  -86.75   <2e-16 ***
puissance12 -3.32324    0.05812  -57.17   <2e-16 ***
puissance13 -3.14545    0.08482  -37.08   <2e-16 ***
puissance14 -3.14705    0.09950  -31.63   <2e-16 ***
puissance15 -3.41704    0.10206  -33.48   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1307478  on 678013  degrees of freedom
Residual deviance:  175926  on 678001  degrees of freedom
AIC: 226763
```

```
Number of Fisher Scoring iterations: 6

> exp(marges$coefficients[6])
puissance9
0.04058501
```

Ce que nous dit la méthode des marges est que cette valeur prédite correspond à la moyenne empirique au sein de la classe de risque,

```
> I=(nombre$puissance=="9")
> mean(nombre$nombre[I])
[1] 0.04058501
```

L'idée est à la fois simple et naturelle sur les modèles ne comportant que des classes de risques (et pas de variable continue).

1.3.3 Prise en compte de l'exposition et variable offset

Dans un modèle collectif, on a besoin de connaître le nombre de sinistres survenus sur une police d'assurance. Dans l'optique de tarifer un contrat, il faut pouvoir prédire le nombre de sinistres qui surviendront, en moyenne, l'année suivante. Or si certains polices n'ont été observées que 6 mois dans la base, il convient de pondérer la fréquence de sinistre par l'exposition. Compte tenu de la propriété multiplicative d'un processus de Poisson, une police observée 1 an aura, en moyenne, 4 fois plus de sinistres qu'une police observée 3 mois. Dans le cas d'un modèle log-Poisson, il est alors naturel de supposer que

$$Y|\boldsymbol{X} \sim \mathcal{P}(\exp[\boldsymbol{X}\boldsymbol{\beta} + \log(e)])$$

où e désigne l'exposition, mesurée en années.

Dans le cas des régressions de Poisson, cela peut se faire de la manière suivante

```
> reg=glm(nombre~0+puissance+region,
+ data=nombre,family=poisson(link="log"),offset=log(exposition))
```

On peut noter que la régression pouvait s'écrire

$$Y|\boldsymbol{X} \sim \mathcal{P}(\exp[\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + e])$$

autrement dit, on rajoute l'exposition dans la régression, tout en forçant le coefficient à être égal à 1. Ceci légitime ainsi la seconde écriture possible

```
> reg=glm(nombre~0+puissance+region+offset(exposition),
+ data=nombre,family=poisson(link="log"))
```

On notera qu'il est possible d'intérgrer une variable offset dans la méthode des marges, en notant qu'il convient de faire une moyenne du nombre de sinistres, divisé par la moyenne de l'exposition. Par exemple pour reprendre une régression présentée en introduction

```
> seuils = c(17,21,25,30,50,80,120)
> reg2 = glm(nombre~cut(ageconducteur,breaks=seuils),data=sinistres,
+ family=poisson(link="log"),offset=log(exposition))
> predict(reg2,newdata=data.frame(ageconducteur=20,exposition=1),
+ type="response")
[1] 0.2113669
> I=(sinistres$ageconducteur>=17)&(sinistres$ageconducteur<=21)
> mean(sinistres$nombre[I==TRUE])/mean(sinistres$exposition[I==TRUE])
[1] 0.2113669
```

Une autre manière d'écrire cette grandeur est de faire une moyenne pondérée (par l'exposition) du nombre annualisé de sinistres,

```
> weighted.mean(sinistres$nombre[I==TRUE]/sinistres$exposition[I==TRUE],
+ w=sinistres$exposition[I==TRUE])
[1] 0.2113669
```

1.3.4 Prise en compte de la surdispersion

Dans une régression poissonnienne, on suppose que dans une classe de risque (ou conditionnellement aux variables explicatives), la fréquence et l'espérance coïncident, i.e. $\text{Var}(Y|\boldsymbol{X}) = \mathbb{E}(Y|\boldsymbol{X})$. Dans l'exemple ci-dessous, on considère le nombre de sinistres RC. On consistue quelques classes tarifaires,

```
> moyenne  = tapply(BASENB$N1RC , BASENB[,c("agecond","carburant,
+ "puissance"")], mean)
> variance = tapply(BASENB$N1RC , BASENB[,c("agecond","carburant,
+ "puissance"")], var)
> taille  = tapply(BASENB$expo , BASENB[,c("agecond","carburant",
+ "puissance")], sum)
> plot(as.vector(moyenne),as.vector(variance))
> abline(a=0,b=1,col="red")
> abline(lm(as.vector(variance)~as.vector(moyenne)),col="blue",lty=2)
```

On peut commencer par faire un premier test, afin de voir si la pente de la régression semble significativement différente

```
> library(AER)
> (regression=lm(as.vector(variance)~as.vector(moyenne),
+  weight=as.vector(taille))
```

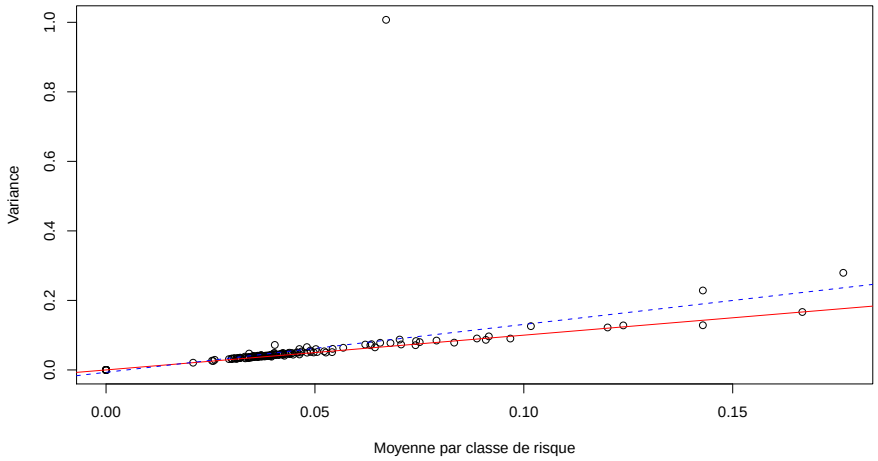


FIGURE 1.21 – Fréquence moyenne et variance à l'intérieur des classes de risques obtenues en segmentant par type de carburant, par puissance et par âge de conducteur. Le trait continu correspond au cas de non-surdisperson ($\mathrm{Var}(Y|\boldsymbol{X}) = \mathbb{E}(Y|\boldsymbol{X})$), et les traits pointillés aux droites de régression (avec ou sans pondération par l'exposition).

```
Call:
lm(formula = as.vector(variance) ~ as.vector(moyenne),
   weights = as.vector(taille))

Coefficients:
       (Intercept)  as.vector(moyenne)
         -0.003966            1.200848
> linear.hypothesis(regression,"as.vector(moyenne)=1")
Linear hypothesis test

Hypothesis:
as.vector(moyenne) = 1

Model 1: as.vector(variance) ~ as.vector(moyenne)
Model 2: restricted model

  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1   1653 67.658
2   1654 70.460 -1   -2.8024 68.468 2.623e-16 ***
---
```

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Une manière de prendre en compte la surdispersion peut être de prendre non une loi de `poisson`, mais une loi `quasipoisson`, telle que $\text{Var}(Y|\boldsymbol{X}) = \phi\mathbb{E}(Y|\boldsymbol{X})$, où ϕ devient un paramètre à estimer (tout comme la volatilité des résidus dans une régression linéaire Gaussienne).

```
> regglm <- glm(N1RC~as.factor(agecond)+carburant+as.factor(puissance),
+ offset=log(expo), data=BASENB,family=quasipoisson)
> summary(regglm)

Call:
glm(formula = N1RC ~ as.factor(agecond) + carburant + as.factor(puissance),
    family = quasipoisson, data = BASENB[I, ], offset = log(expo))

(Dispersion parameter for quasipoisson family taken to be 1.583862)

> (summary(regglm)$dispersion)
[1] 1.583862
```

Pour tester la présence d'une éventuelle surdispersion, on peut noter que la surdispersion correspond à une hétérogénéité résiduelle, c'est à dire un effet aléatoire. Par exemple on peut supposer que

$$(Y|\boldsymbol{X}=\boldsymbol{X},\boldsymbol{Z}=\boldsymbol{z}) \sim \mathcal{P}(\exp[\boldsymbol{X}'\boldsymbol{\beta}+\boldsymbol{z}'\boldsymbol{\alpha}])$$

de telle sorte que si $u = \boldsymbol{z}'\boldsymbol{\alpha} - \mathbb{E}(\boldsymbol{Z}'\boldsymbol{\alpha}|\boldsymbol{X}=\boldsymbol{X})$, alors

$$(Y|\boldsymbol{X}=\boldsymbol{X},\boldsymbol{Z}=\boldsymbol{z}) \sim \mathcal{P}(\exp[\boldsymbol{X}'\boldsymbol{\gamma}+u])$$

On a un modèle dit à effets fixes, au sens où

$$(Y|\boldsymbol{X}=\boldsymbol{X}) \sim \mathcal{P}(\exp[\boldsymbol{X}'\boldsymbol{\gamma}+U])$$

où $U = \boldsymbol{Z}'\boldsymbol{\alpha} - \mathbb{E}(\boldsymbol{Z}'\boldsymbol{\alpha}|\boldsymbol{X}=\boldsymbol{X})$. Par exemple, si on suppose que $U \sim \gamma(a,a)$, i.e. d'espérance 1 et de variance $\sigma^2 = 1/a$, alors

$$(Y|U=u) \sim \mathcal{P}(\lambda u) \text{ où } \lambda = \exp[\boldsymbol{X}'\boldsymbol{\gamma}]$$

de telle sorte que

$$\mathbb{E}(Y|U=u) = \text{Var}(Y|U=u).$$

Mais si on regarde la loi nonconditionnelle, $\mathbb{E}(Y) = \lambda$ alors que

$$\text{Var}(Y) = \text{Var}(\mathbb{E}[Y|U]) + \mathbb{E}(\text{Var}(Y|)) = \lambda + \lambda^2\sigma^2.$$

On peut alors proposer un test de la forme suivante : on suppose que

$$\mathrm{Var}(Y|\boldsymbol{X}=\boldsymbol{X}) = \mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{X}) + \tau \cdot \mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{X})^2$$

on on cherche à tester

$$H_0 : \tau = 0 \text{ contre } \tau > 0$$

Parmi les statistiques de test classique, on pourra considérer

$$T = \frac{\sum_{i=1}^n [(Y_i - \widehat{\mu}_i)^2 - Y_i]}{\sqrt{2\sum_{i=1}^n \widehat{\mu}_i^2}}$$

qui suit, sous H_0, une loi normale centrée réduite. On utilise simplement `dispersiontest()` de `library(MASS)`.

```
> regpoisson=glm(N1RC~as.factor(agecond)+carburant+as.factor(puissance),offset=lo
+ data=BASENB,family=poisson)
> dispersiontest(regpoisson)

        Overdispersion test

data:  regpoisson
z = 6.4039, p-value = 7.572e-11
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
  1.069558
```

Une autre possibilité est de faire une régression binomiale négative (qui permettra de prendre en compte de la *sur*dispersion). Elle se fait à l'aide de la fonction `glm.nb()` de `library(MASS)`.

```
> regbn <- glm.nb(N1RC~as.factor(agecond)+carburant+as.factor(puissance)
+ offset(log(expo)),data=BASENB[I,])
```

Remarque 1.3 *La loi Binomial Négative est obtenue comme un mélange Poisson-Gamma. Dans* `library(gamlss)` *on parle de loi binomiale négative de type I. Une loi de type II est obtenue en considérant un mélange Poisson-inverse Gaussienne.*

1.3.5 Les modèles *zero-inflated*

Afin d'éviter l'aléa moral, il n'est pas rare de mettre en place des contrats participatifs. En assurance, l'exemple le plus connu est probablement le mécanisme de bonus-malus. Une personne qui n'a pas d'accident responsable

une année a le droit à un rabais l'année suivante (un *bonus*) alors qu'une personne ayant eu un ou plusieurs sinistres subit une majoration de prime (un *malus*). D'un point de vue économétrique, cette solution présente un biais puisqu'elle peut insiter des personnes à ne pas déclarer certains sinistres (dès lors que la majoration excède le coût du sinistre). Il n'est alors pas rare d'observer *trop* de personnes non-sinistrées dans la population totale (par rappport à un modèle Poissonnien).

Un modèle dit *zero inflated* est un mélange entre une masse en 0 et un modèle classique de comptage, typiquement un modèle de Poisson, ou binomial négatif. Pour modéliser la probabilité de ne pas déclarer un sinistre (et donc d'avoir un *sur*poids en 0), considérons un modèle logistique par exemple,

$$\pi_i = \frac{\exp[\boldsymbol{X}_i'\boldsymbol{\beta}]}{1+\exp[\boldsymbol{X}_i'\boldsymbol{\beta}]}$$

Pour le modèle de comptable, on note $p_i(k)$ la probabilité que l'individu i ait k sinistres. Aussi,

$$\mathbb{P}(N_i=k) = \begin{cases} \pi_i + [1-\pi_i]\cdot p_i(0) \text{ si } k=0, \\ [1-\pi_i]\cdot p_i(k) \text{ si } k=1,2,\cdots \end{cases}$$

Si p_i correspond à un modèle Poissonnien, on peut alors montrer facilement que $\mathbb{E}(N_i)=[1-\pi_i]\mu_i$ et $\text{Var}(N_i)=\pi_i\mu_i+\pi_i\mu_i^2[1-\pi_i]$.

`library(gamlss)` propose la fonction `ZIP` (pour *zero inflated Poisson*), mais aussi `ZINBI` (lorsque p_i correspond à une loi binomiale négative), ou `ZIPIG` (pour un mélange Poisson-inverse Gaussien), par exemple. La `library(pscl)` propose également une fonction `zeroinfl` plus simple d'utilisation, proposant aussi bien un modèle de Poisson qu'un modèle binomial négatif.

Il existe aussi des modèles dits *zero adapted*, où l'on suppose que

$$\mathbb{P}(N_i=k) = \begin{cases} \pi_i \text{ si } k=0, \\ [1-\pi_i]\cdot\dfrac{p_i(k)}{1-p_i(0)} \text{ si } k=1,2,\cdots \end{cases}$$

Dans `library(gamlss)` il s'agit du modèle `ZAP`. Comme auparavant, il existe des fonctions `ZANBI` ou `ZAPIG`.

Ces modèles à inflation zéro peuvent être particulièrement utiles pour prendre en compte un excès de non-déclarations de sinistres, généralement attribuées à une peur de perdre un niveau intéressant de bonus-malus : la perte financière associée au malus des années suivantes peut excéder l'indemnité versée aujourd'hui. On peut ajuster ici un modèle zero-inflated (logit) avec une loi de Poisson afin d'expliquer la sinistralité en fonction de l'âge du conducteur (en prenant en compte l'âge via une fonction nonlinéaire que l'on estimera à l'aide de splines).

```
> reg1 <- glm(nombre~ageconducteur,offset=exposition,data=nombre,
+ family=poisson)
> reg2 <- glm(nombre~bs(ageconducteur,df=4),offset=exposition,
+ data=nombre,family=poisson)
> reg3 <- zeroinfl(nombre~ageconducteur | ageconducteur,
+ data = nombre,offset=exposition,dist = "poisson",link="logit")
> reg4 <- zeroinfl(nombre~bs(ageconducteur,df=4) | bs(ageconducteur),
+ data = nombre,dist = "poisson",link="logit",offset=exposition)
```

La prédiction obtenue pour les âges usuels est présentée sur la Figure 1.22. Si l'on ne prend pas en compte l'âge de manière nonlinéaire, les deux modèles prédisent sensiblement la même chose.

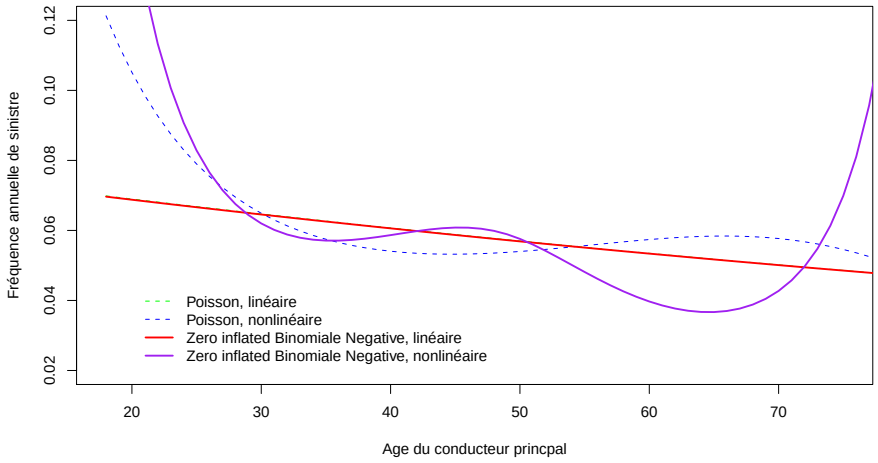


FIGURE 1.22 – Prédiction à l'aide de modèles zero-inflated (logit) avec une loi de Poisson de la sinistralité en fonction de l'âge du conducteur.

On peut s'intéresser plus particulièrement à l'impact de l'âge sur la probabilité de ne pas déclarer de sinistres (correspondant au paramètre de la loi binomiale).

```
> age=seq(18,80)
> DT=data.frame(ageconducteur=age,exposition=1)
> Y4z <- predict(reg4,newdata=DT,type="zero")
> plot(age,Y4z)
```

On notera que l'interprétation en terme de niveau de bonus-malus semble pertinente, en particulier si l'on regarde le *bonus moyen* en fonction de l'âge du conducteur, présenté sur la Figure 1.24 : le taux de bonus (et donc la

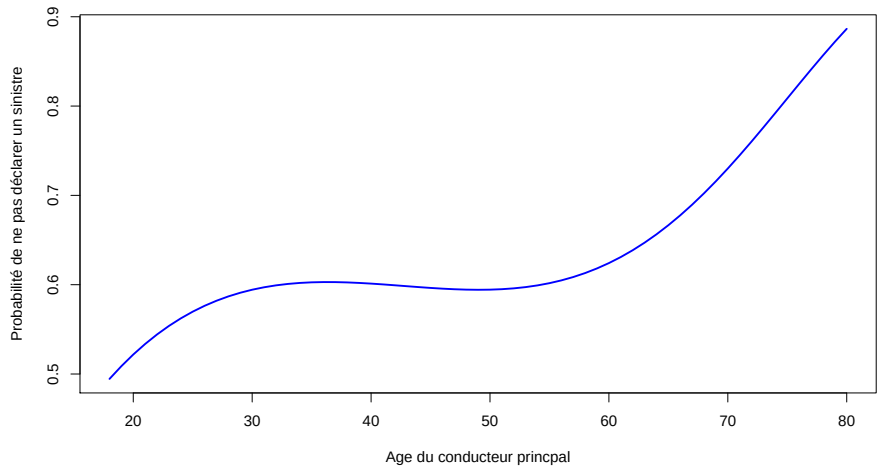


FIGURE 1.23 – Probabilité π_i du modèle à inflation zéro, interprétée comme la probabilité de ne pas déclarer un sinistre, en fonction de l'âge du conducteur.

prime) diminue avec l'âge, ce qui incite probablement à ne pas déclarer certains petits sinistres responsables.

1.3.6 Régression simple versus régression multiple

Il est important de bien vérifier les interractions entre les variables explicatifs dans la régression, afin d'être certain que l'effet est bien additif.

1.3.7 Prédiction de la fréquence par police

Nous avons vu qu'il était possible d'ajuster un grand nombre de modèles, en changeant la *loi* (Poisson, zero-inflated Poisson, Binomiale Négative, Poisson-inverse Gaussienne) de la variable N, mais aussi la forme du modèle (l'âge intervenant tel quel, par classe, ou bien transformée de manière nonlinéaire, par exemple). Une cinquantaine de modèles ont été ajustés. Afin de comparer ces modèles, on calcule le critère d'Akaike (AIC) ou de Schwarz (BIC). On peut aussi prédire la fréquence pour quelques individus type,

```
> individus=data.frame(
+ exposition=c(1,1,1),
+ zone=c("B","C","D"),
+ puissance=c(11,6,7),
+ agevehicule=c(0,3,10),
+ ageconducteur=c(40,18,65),
```

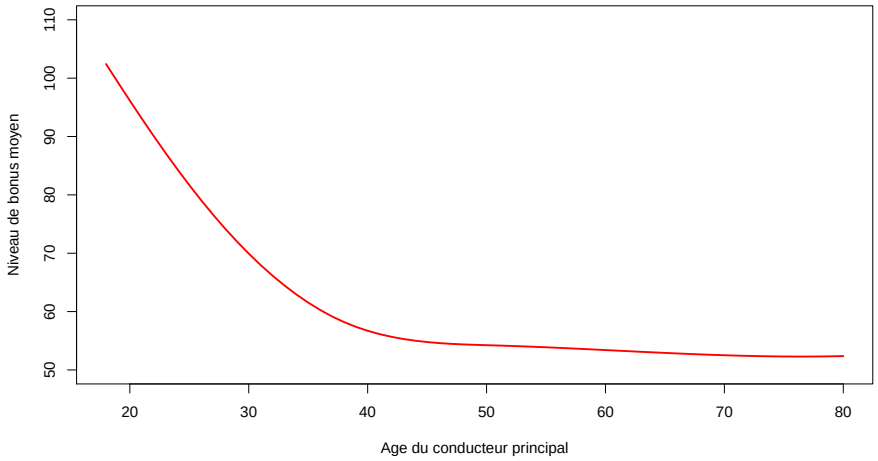


FIGURE 1.24 – Niveau moyen de taux de bonus en fonction de l'âge du conducteur.

```
+ marque=c(1,2,10),
+ carburant=c("D","E","D"),
+ densite=c(11,24,93),
+ region=c(13,13,7))
> individus
  exposition zone puissance agevehicule
1          1    B        11           0
2          1    C         6           3
3          1    D         7          10
  ageconducteur marque carburant densite region
1            40      1         D      11     13
2            18      2         E      24     13
3            65     10         D      93      7
```

Il est aussi possible d'utiliser les arbres de régression afin de mieux comprendre les différences entre les modèles.

```
> I=sample(1:nrow(sinistres),size=150000,replace=FALSE)
> base1=sinistres[I,]
> base2=sinistres[-I,]
> reg1 = glm(nombre~puissance+agevehicule+ageconducteur+carburant+as.factor(regio
+ data=base1,family=poisson,offset=log(exposition))
> library(mgcv)
> reg2 = gam(nombre~zone+puissance+s(agevehicule)+s(ageconducteur)+carburant+as.f
```

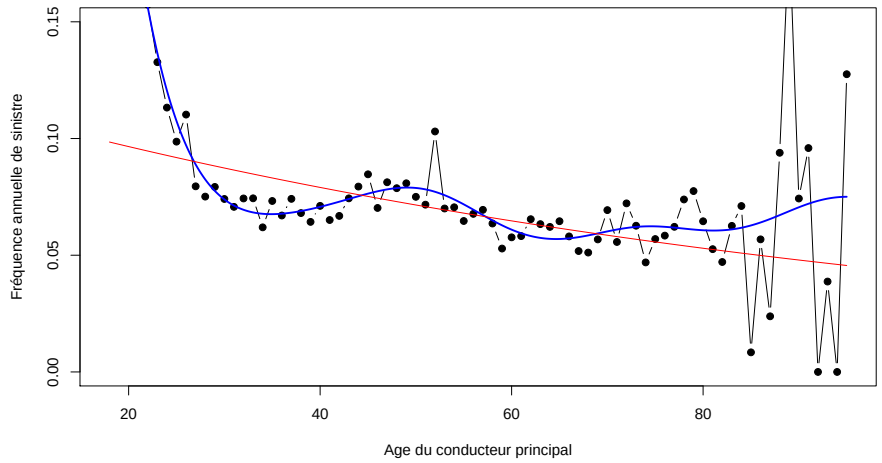


FIGURE 1.25 – Fréquence individuelle en fonction de l'âge du conducteur principal.

```
+ data=base1,family=poisson,offset=log(exposition))
> Y1=predict(reg1,newdata=base2)
> Y2=predict(reg2,newdata=base2)
> diff=(Y1-Y2)
> base2$INDIC = (diff>0)
> base2$diff = diff
> library(tree)
> arbre <- tree(diff~puissance+agevehicule+ageconducteur+carburant+zone+densite,data=base
> plot(arbre)
> text(arbre)
```

On cherche à comparer sur quelles segments de population les modèles donnent des prédictions sensiblement différentes. Les deux modèles ont la même stucture (log-Poisson), mais le premier ne prend pas en compte les aspects nonlinéaires. Le premier n'intègre pas non plus la variable de densité de populaion. Visiblement, il y a une forte différence de prédiction

- sur les régions 'faiblement' peuplées, `densite<482.5`
- sur les jeunes conducteurs, `ageconducteur<24.5`
- sur les véhicules neufs, `agevehicule<0.5`

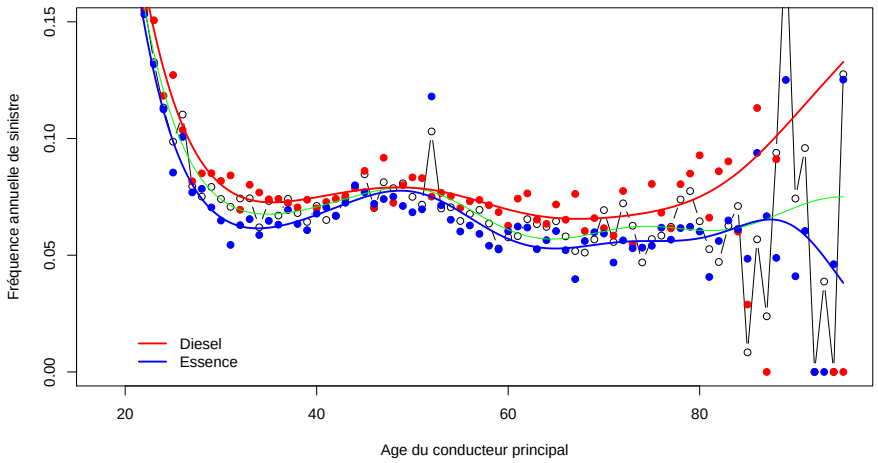


FIGURE 1.26 – Fréquence individuelle en fonction de l'âge du conducteur principal, et du type de carburant.

1.4 Modéliser les coûts individuels des sinistres

Les coûts de sinistres sont des variables positives. En présence de coûts fixes (bris de glace par exemple), la loi des coûts de sinistres sera une loi continue, avec des masses de Dirac (là on l'on observe des coûts fixes). La loi est alors

$$f(y) = (1-p)f_\star(y) + p\mathbf{1}(y = C)$$

où p désigne la probabilité d'avoir un coût qui soit précisément C. Dans notre approche économétrique, on peut envisager un modèle de la forme

$$f(y|\boldsymbol{X} = \boldsymbol{x}) = (1-p(\boldsymbol{x}))f_\star(y|\boldsymbol{X} = \boldsymbol{x}) + p(\boldsymbol{x})\mathbf{1}(y = C)$$

où $p(\boldsymbol{x})$ peut être modélisée par une régression logistique, et où $f_\star(y|\boldsymbol{X} = \boldsymbol{x})$ est une loi positive *à densité*.

On peut alors chercher à modéliser cette loi continue.

1.4.1 Modèle Gamma et modèle lognormal

Les deux modèles les plus classiques permettant de modéliser les coûts individuels de sinistre sont

- le modèle Gamma sur les coûts individuels Y_i
- le modèle log-normal sur les coûts individuels Y_i, ou plutôt un modèle Gaussien sur le logarithme des coûts, $\log(Y_i)$, la loi lognormale n'appartenant pas à la famille exponentielle.

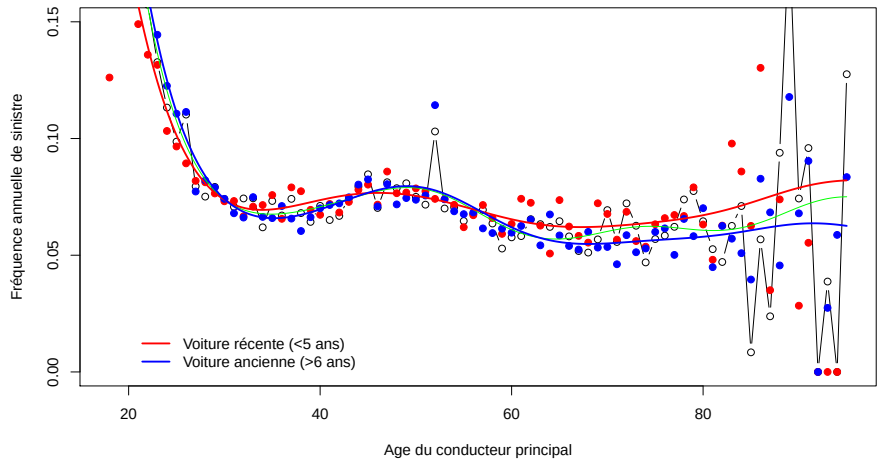


FIGURE 1.27 – Fréquence individuelle en fonction de l'âge du conducteur principal, et de l'ancienneté du véhicule.

Le(s) modèle(s) Gamma

La loi Gamma, de paramètres α et β, de densité

$$f(y) = \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} \exp(-\beta y), \text{ pour } y \geq 0,$$

vérifie $\mathbb{E}(Y) = \frac{\alpha}{\beta}$ et $\text{Var}(X) = \frac{\alpha}{\beta}^2$. Autrement dit, le coefficient de variation vaut ici

$$\text{CV} = \frac{\sqrt{\text{Var}(X)}}{\mathbb{E}(Y)} = \frac{1}{\sqrt{\alpha}}$$

qui peut être analysé comme un cofficient de *dispersion*. En fait, si $\phi = 1/\alpha$, on peut écrire

$$\text{Var}(Y) = \frac{1}{\alpha}\frac{^2}{\beta^2} = \phi \cdot \mathbb{E}(Y)^2,$$

où on retrouve ici une fonction variance de forme quadratique.

Le cas particulier $\phi = 1$ corrrespond à la loi *exponentielle*.

Bien que le lien canonique de la loi Gamma soit la fonction inverse, il est plus fréquent d'utiliser un lien logarithmique. En effet, la forme multiplicative donne des interprétations simples dans le cas des modèles multiples.

```
> reggamma <- glm(cout~ageconducteur,family=Gamma(link="log"),
+ data=sinistres)
> summary(reggamma)
```

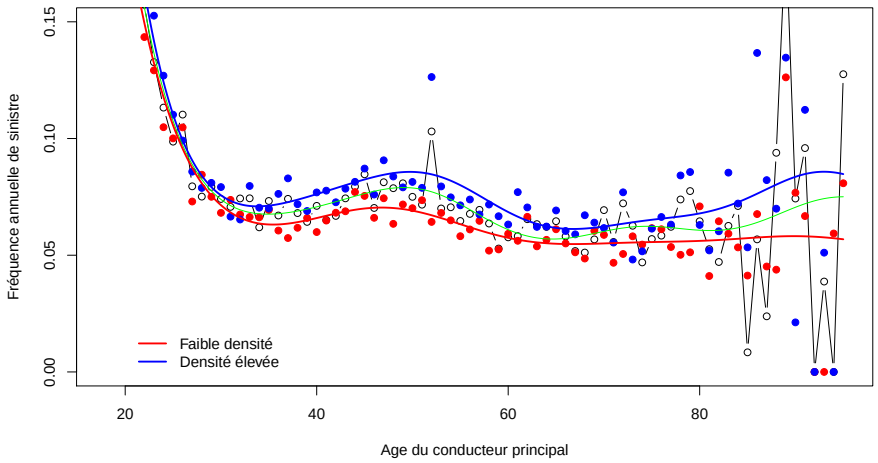


FIGURE 1.28 – Fréquence individuelle en fonction de l'âge du conducteur principal, et de la densité de la région d'habitation.

```
Call:
glm(formula = cout ~ ageconducteur, family = Gamma(link = "log"),
    data = sinistres)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6322  -0.9769  -0.6110  -0.3917  52.5993

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    8.180643   0.208009  39.328   <2e-16 ***
ageconducteur -0.010440   0.004383  -2.382   0.0172 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for Gamma family taken to be 109.7107)

    Null deviance: 46482  on 26443  degrees of freedom
Residual deviance: 45637  on 26442  degrees of freedom
AIC: 458704

Number of Fisher Scoring iterations: 9
```

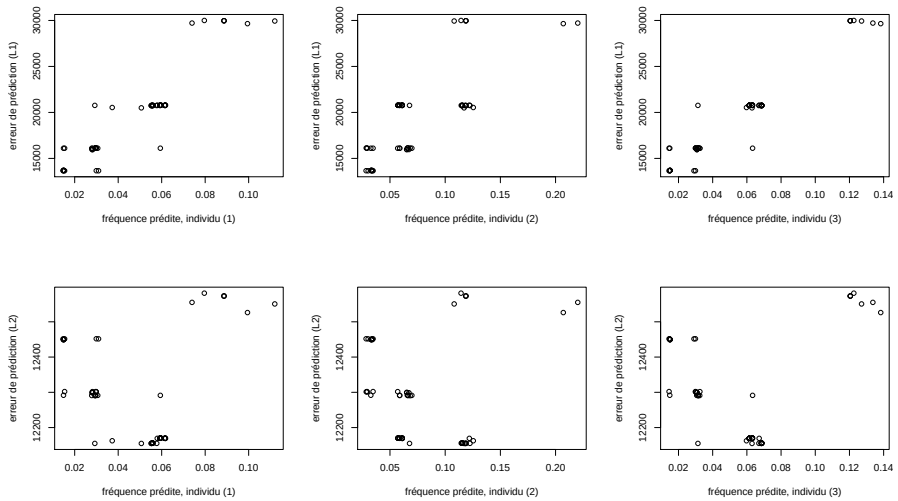


FIGURE 1.29 – Estimation de $\mathbb{E}(N|\boldsymbol{X})$ sur 3 individus type, à l'aide d'une cinquantaine de modèle.

Si on s'intéresse à la valeur prédite pour une personne d'âge `ageconducteur=50`, on obtient

```
> predict(reggamma,newdata=data.frame(ageconducteur=50),
+ type="response")
       1
2118.879
```

Le modèle lognormal

La régression lognormale peut être obtenue en considérant une régression linéaire (Gaussienne) sur le logarithme du coût,

$$\log(Y_i) = \boldsymbol{X}_i'\boldsymbol{\beta} + \varepsilon_i$$

avec $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. En effet, par définition de la loi lognormale, $Y \sim LN(\mu, \sigma^2)$ si et seulement si $\log Y \sim \mathcal{N}(\mu, \sigma^2)$. Le principal soucis dans cet écriture est que

$$\begin{cases} \mathbb{E}(Y) = \exp\left(\mu + \dfrac{\sigma^2}{2}\right) \neq \exp(\mu) = \exp[\mathbb{E}(\log Y)] \\ \mathrm{Var}(Y) = \exp\left(2\mu + \sigma^2\right)\left[\exp\left(\sigma^2\right) - 1\right] \neq \exp(\sigma^2) = \exp[\mathrm{Var}(\log Y)] \end{cases}$$

Autrement dit, il sera délicat de passer des estimations faites à partir du modèle sur $\log Y$ à des prédictions sur le coût Y. Une régression sur le logarithme des coûts donnerait par exemple,

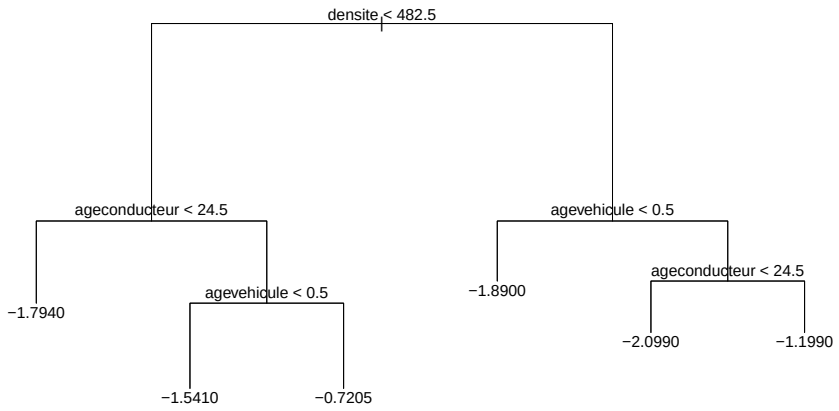


FIGURE 1.30 – Comparaison de deux modèles log-Poisson.

```
> reglm <- lm(log(cout)~ageconducteur,data=sinistres)
> summary(reglm)

Call:
lm(formula = log(cout) ~ ageconducteur, data = sinistres)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.7501521  0.0224328 300.905  < 2e-16 ***
ageconducteur 0.0021392  0.0004727   4.525 6.06e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1.13 on 26442 degrees of freedom
Multiple R-squared: 0.0007738,  Adjusted R-squared: 0.0007361
F-statistic: 20.48 on 1 and 26442 DF,  p-value: 6.059e-06
> sigma=summary(reglm)$sigma
```

Si on s'intéresse à la valeur prédite pour une personne d'âge `ageconducteur=50`, on obtient

```
> mu=predict(reglm,newdata=data.frame(ageconducteur=50))
> exp(mu+sigma^2/2)
       1
1799.239
```

On notera que les deux modèles donnent des résultats *très* sensiblement différents (en terme de signe par exemple). On peut comparer les prédictions sur la Figure 1.31 (sur laquelle des régressions nonparamétriques ont été superposées).

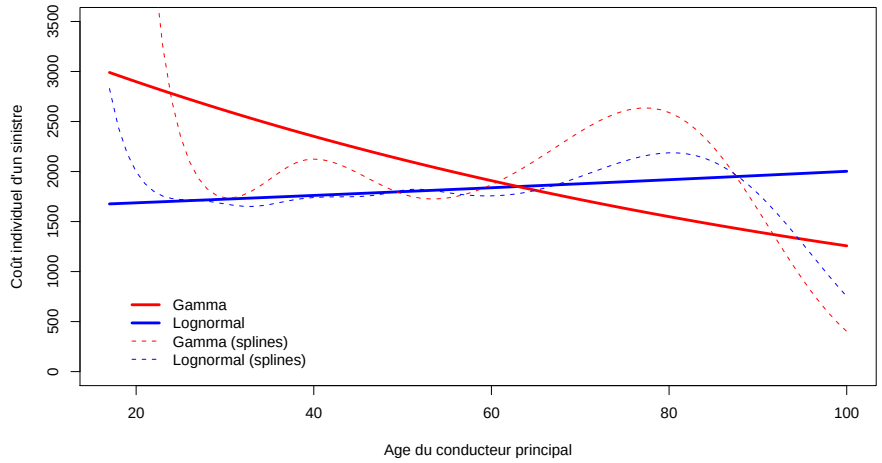


FIGURE 1.31 – Régressions lognormale versus Gamma, où le coût individuel est expliqué par l'âge du conducteur.

La Figure 1.32 montre les mêmes types de modèles si l'on cherche à expliquer le coût par l'ancienneté du véhicule. En particulier, la croissance du coût moyen en fonction de l'âge du véhicule est surprenante compte tenu de la baisse de la cote du véhciule à l'argus,

En fait, la divergence entre les deux modèles vient du fait que le modèle Gamma est très sensible aux valeurs extrêmes. Un avantage du modèle lognormal est qu'en prenant le logarithme des coûts, on atténue l'importance des sinistres de coût exceptionnel. En écartant les sinistres tels que `sinistres$cout>200000`, on obtient des modèles comparables (et proches de ce que donnait la régression lognormale sur l'ensemble de la base)

```
> sinistrescap <- sinistres[sinistres$cout<200000,]
```

Nous reviendrons plus en détails sur la prise en compte de ces sinistres exceptionnels (qui ici ont simplement été écartés).

Prise en compte d'un montant maximal

Dans la plupart des assurances associées aux dommages matériels, les polices indiquent des montants maximals. Dans le cas où seul le véhicule

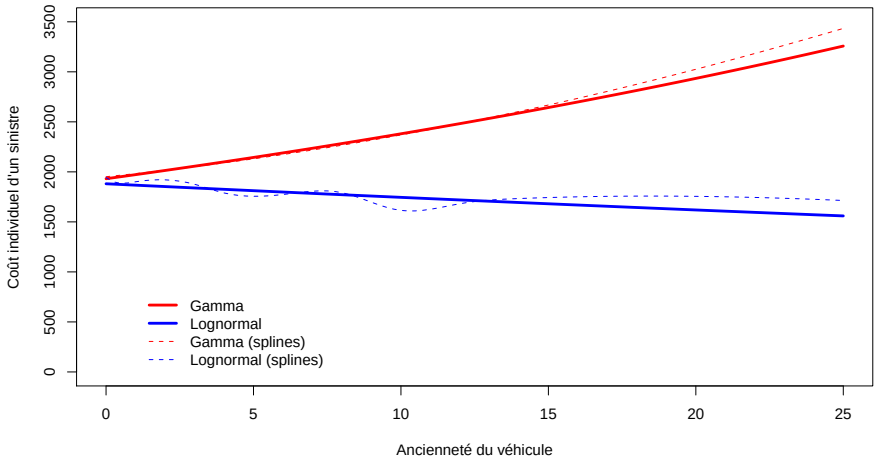


FIGURE 1.32 – Régressions lognormale versus Gamma, où le coût individuel est expliqué par l'ancienneté du véhicule.

de l'assuré est couvert, le montant maximal d'un sinistre matériel est la valeur du véhicule. Dans la garantie vol des contrats habitations, le montant maximal sera la valeur des biens assurés. Y est ainsi une variable censurée par ce coût maximal (qui peut lié à la police).

Les `library(Zelig)` et `library(HMisc)` permettent de faire une régression lognormale, dans le contexte des modèles de durée. Pour utiliser la fonction `library(Zelig)` on va indiquer que tous les coûts sont noncensurés (car ici on ne dispose pas de l'information)

```
> library(Zelig)
> regloglm <- zelig(Surv(cout, rep(1,length(cout))) ~ ageconducteur,
+ model = "lognorm", data = sinistres)
> summary(regloglm)

Call:
zelig(formula = Surv(cout, rep(1, length(cout))) ~ ageconducteur,
    model = "lognorm", data = sinistres)
               Value Std. Error      z         p
(Intercept)   6.75015   0.022432 300.92  0.00e+00
ageconducteur 0.00214   0.000473   4.53  6.03e-06
Log(scale)    0.12183   0.004348  28.02 9.76e-173

Scale= 1.13
```

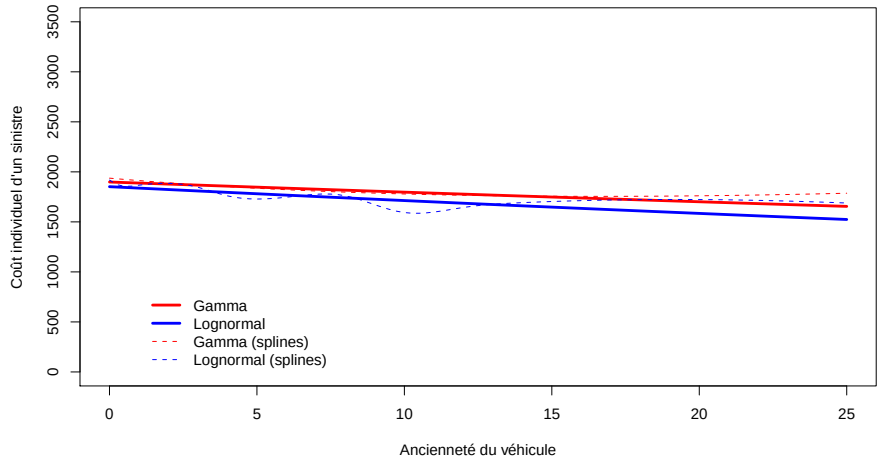


FIGURE 1.33 – Régressions lognormale versus Gamma, où le coût individuel est expliqué par l'ancienneté du véhicule.

```
Log Normal distribution
Loglik(model)= -221797.7   Loglik(intercept only)= -221807.9
        Chisq= 20.47 on 1 degrees of freedom, p= 6.1e-06
Number of Newton-Raphson Iterations: 2
n= 26444
```

1.4.2 Modélisation des grands sinistres

Il existe un grand nombre de façons de définir les lois à queues épaisses. La plus élégante d'un point de vue actuarielle est probablement la famille des lois *sous exponentielles* (décrites dans [?]). Une loi de fonction de survie $\overline{F}$ sera dite sous-exponentielle si pour tout $n \geq 2$,

$$\lim_{x\to\infty} \frac{\overline{F^{\star n}}(x)}{\overline{F}(x)} = n$$

ou bien, si $X_1, \cdots, X_n, \cdots$ sont des variables i.i.d. de loi F,

$$\mathbb{P}(X_1 + \cdots + X_n > x) \sim \mathbb{P}(\max\{X_1, \cdots, X_n\} > x).$$

Autrement dit, la loi de la charge totale dans un portefeuille a des queues des distributions qui se comportent comme le plus gros sinistres. Ce sont donc des lois qui sont très influencées par ces très gros sinistres. Parmi les lois de la famille sous-exponentielle,

– la loi lognormale, $f(y) \propto \dfrac{1}{y\sigma} \exp\left(-[\log y - \mu]^2/2\sigma^2\right)$

– la loi de Weibull, $f(y) \propto x^{k-1}\exp[-x^k]$ si $k<1$

mais la loi la plus utilisée, en particulier en réassurance, n'est pas dans la famille exponentielle,

– la loi de Pareto, $f(y) \propto [\mu+y]^{-\alpha-1}$

Dans ces familles de lois à queues épaisses, on va ainsi retrouver une autre classe relativement connue, à savoir les lois dite *à variation régulière*. Ces lois sont aussi dite *de type* Pareto, au sens où

$$\mathbb{P}(Y>y) = y^{-\alpha}\mathcal{L}(y)$$

où $\mathcal{L}$ est une fonction à variation lente, i.e.

$$\lim_{x\to\infty}\frac{\mathcal{L}(tx)}{\mathcal{L}(x)} = 1 \text{ pour tout } t>0.$$

La `library(gamlss)` propose d'autres familles de lois, comme les lois *Reverse Gumbel* ou *Power Exponential*

Il est possible de définir une famille dite *beta généralisée de seconde espèce*, notée GB2. On suppose que

$$\log Y \overset{\mathcal{L}}{=} \mu + \sigma \log\frac{\Gamma_1}{\Gamma_2}$$

où $\Gamma\sim\mathcal{G}(\alpha_i,1)$ sont indépendantes. Si Γ_2 est une constante ($\alpha_2\to\infty$) on obtient la *loi gamma généralisée*.

La densité de cette loi s'écrit :

$$f(y)\propto y^{-1}\left[\exp\left(\frac{\log y-\mu}{\sigma}\right)\right]^{\alpha_1}\left[1+\exp\left(\frac{\log y-\mu}{\sigma}\right)\right]^{-(\alpha_1+\alpha_2)}$$

Supposons que μ soit une fonction linéaire des variables explicatives, $\mu=\boldsymbol{X}'\boldsymbol{\beta}$. Alors

$$\mathbb{E}(Y|\boldsymbol{X}) = C\exp[\mu(\boldsymbol{X})] = C\exp[\boldsymbol{X}'\boldsymbol{\beta}]$$

Ces modèles sont détaillés dans [23].

1.4.3 Ecrêtement des grands sinistres

Si l'on considère des modèles économétriques basés uniquement sur des variables catégorielles (en particulier des classes pour les variables continues) la prime pure est alors généralement la moyenne empirique dans la classe considérée (c'est en tous les cas ce que préconise par exemple la méthode des marges). Mais cette méthode devient alors vite très sensible aux sinistres extrêmes.

Afin d'éviter ce problème, il n'est pas rare d'écrêter les sinistres : on calcule la prime moyenne par groupe tarifaire en écartant les gros sinistres, qui

seront répartis sur l'ensemble de la population. On peut bien entendu raffiner cette méthode en considérant des modèles hiérarchiques et en répartissant simplement sur une surclasse.

Supposons que les sinistres extrêmes soient ceux qui dépassent un seuil s (connu). Rappelons que la formule des probabilités totales permet d'écrire que (dans le cas discret pour faire simple)

$$\mathbb{P}(A) = \sum_i \mathbb{P}(A \cap B_i) \sum_i \mathbb{P}(A|B_i) \cdot \mathbb{P}(B_i)$$

où (B_i) forme une partition de Ω. En particulier

$$\mathbb{P}(A) = \mathbb{P}(A|B) \cdot \mathbb{P}(B) + \mathbb{P}(A|B^c) \cdot \mathbb{P}(B^c)$$

où B^c désigne le complémentaire de B. En passant à l'espérance, et en travaillant sur des variables aléatoires plutôt que des ensembles, on peut écrire

$$\mathbb{E}(Y) = \mathbb{E}(Y|B) \cdot \mathbb{P}(B) + \mathbb{E}(Y|B^c) \cdot \mathbb{P}(B^c)$$

Si on prend comme cas particulier $B = \{Y \leq s\}$ et $B^c = \{Y > s\}$, alors

$$\mathbb{E}(Y) = \mathbb{E}(Y|Y \leq s) \cdot \mathbb{P}(Y \leq s) + \mathbb{E}(Y|Y > s) \cdot \mathbb{P}(Y > s).$$

finallement, on note que la probabilité $\mathbb{P}$ n'a joué aucun rôle ici, et on peut parfaitement la remplacer par une probabilité conditionnelle, $\mathbb{P}_{\mathbb{X}}$, i.e.

$$\mathbb{E}(Y|\boldsymbol{X}) = \mathbb{E}(Y|\boldsymbol{X}, Y \leq s) \cdot \mathbb{P}(Y \leq s|\boldsymbol{X}) + \mathbb{E}(Y|\boldsymbol{X}, Y > s) \cdot \mathbb{P}(Y > s|\boldsymbol{X})$$

Le premier terme correspond aux sinistres 'normaux' par une loi évoquée précédemment (régression Gamma par exemple). Pour le second terme, on noteta que $\mathbb{E}[\mathbb{E}(Y|\boldsymbol{X}, Y > s)] = \mathbb{E}(Y|Y > s)$. Autrement dit, on peut être tenté par ne plus distinguer par classe pour le coût moyen des très très gros sinistres. Mais on répartira proportionnellement à la fréquence des gros sinistres sinistres.

La prédiction sera donc basée sur trois parties, la première pour les sinistres usuels (plus petits que s), et la seconde pour les grands sinistres (pour les sinistres excédant s), avec comme troisième terme que sera la probabilité, par classe tarifaire, d'avoir un sinistre excédant le seuil s.

```
> seuil=500000
> sinistres.inf = sinistres[sinistres$cout<=seuil,]
> sinistres.sup = sinistres[sinistres$cout>seuil,]
> sinistres$indic = sinistres$cout>seuil
> proba=gam(indic~s(ageconducteur),data=sinistres,family=binomial)
> probpred=predict(proba,newdata=data.frame(ageconducteur=age),type="response")
> reg=gam(cout~s(ageconducteur),data=sinistres.inf,family=Gamma(link="log"))
> Y.inf=predict(reg,newdata=data.frame(ageconducteur=age),type="response")
> Y.sup=mean(sinistres.sup$cout)
> Y=Y.inf*(1-probpred)+Y.sup*probpred
> plot(age,Y)
```

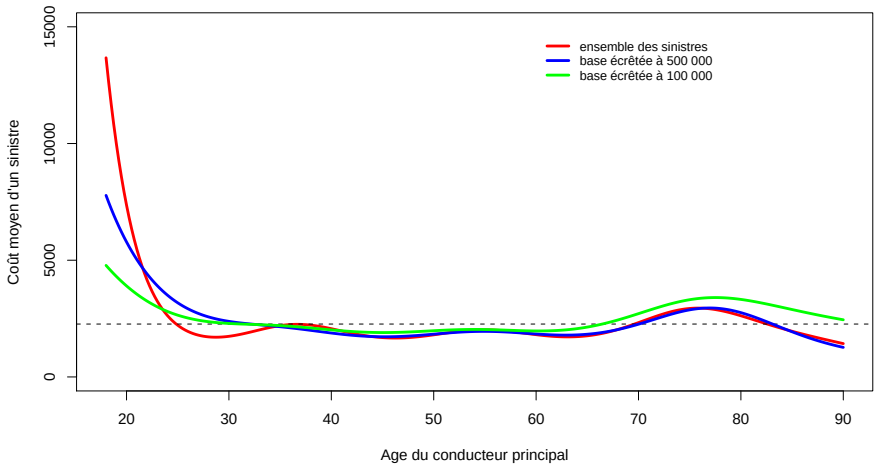


FIGURE 1.34 – Estimation de $\mathbb{E}(Y|\boldsymbol{X})$ avec ou sans écrêment (la surcrête ici ici répartie entre les assurés proportionnellement à leur probabilité d'avoir un gros sinistre).

`probpred` est ici la prédiction de $\mathbb{P}(Y > s|\boldsymbol{X})$. La figure 1.35 montre comment la charge surcrête est répartie entre les assurés : pour un seuil s à 500 000 euros, les très jeunes conducteurs (moins de 22 ans) paieront moins, contrairement aux assurés de 25 à 35 ans.

1.5 Modéliser les coûts par police

Dans certains cas, on ne dispose que de la charge totale annuelle par police d'assurance.

1.5.1 Les modèles Tweedie comme modèle Poisson composé

Les modèles Tweedie peuvent être vu comme des modèles Poisson composés. On suppose que

$$Y = \sum_{k=0}^{N} Z_k$$

où les (Z_k) sont i.i.d., on on pourra supposer qu'ils suivent une loi Gamma $\mathcal{G}(\alpha, \beta)$, indépendament de $N \sim \mathcal{P}(\lambda)$. Alors

$$\mathbb{E}(Y) = \mathbb{E}(N) \cdot \mathbb{E}(Z_k) = \lambda \frac{\alpha}{\beta} = \mu$$

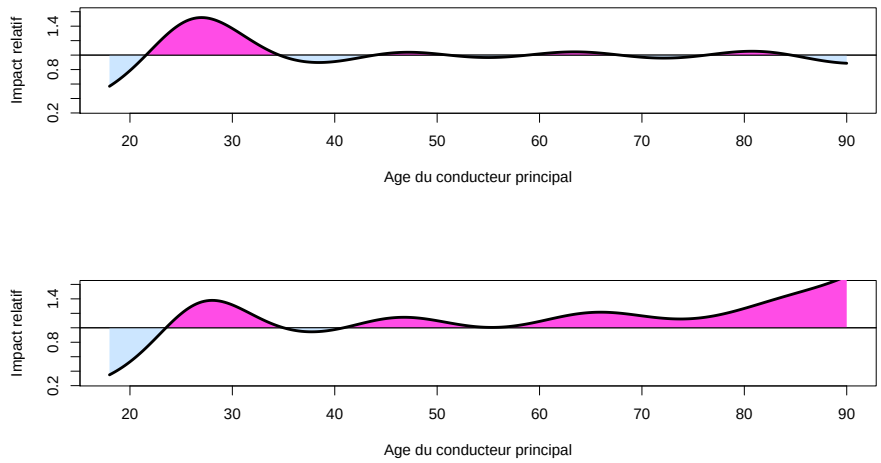


FIGURE 1.35 – Impact (relatif) de l'écrêtement, pour un seuil à 100 000 euros en haut, et 500 000 en bas.

et

$$\mathrm{Var}(Y) = \mathbb{E}(N) \cdot \mathbb{E}(Z_k^2) + \mathrm{Var}(N) \cdot \mathbb{E}(Z_k)^2 = \lambda \left(\frac{\alpha}{\beta^2} + \frac{\alpha^2}{\beta^2} \right)$$

Supposons qu'il existe $p \in]1, 2[$ et $\psi > 0$ tels que

$$\alpha = \frac{2-p}{p-1}, \beta = \frac{1}{\psi(p-1)\mu^{p-1}} \text{ et } \lambda = \frac{\mu^{2-p}}{\psi(2-p)}$$

alors on peut montrer que la loi de Y appartient à la famille exponentielle avec

$$\mathbb{E}(Y) = \mu \text{ et } \mathrm{Var}(Y) = \psi\mu^p$$

où le paramètre ψ est un paramètre de dispersion, et la fonction variance est alors $V(\mu) = \mu^p$.

Afin de mettre en oeuvre l'utilisation de ces modèles, commençons par sommer les coûts de sinistres par police.

```
>  base=sinistres[,1:2]
>  head(base)
  nocontrat    cout
1       139  303.00
2       190 1981.84
3       414 1456.55
4       424  989.64
5       424 9844.36
```

```
6       463 3986.67
>  somme <- aggregate(base[,2],by=list(base[,1]),sum)
>  names(somme)=c("nocontrat","cout")
>  head(somme)
  nocontrat     cout
1       139   303.00
2       190  1981.84
3       414  1456.55
4       424 10834.00
5       463  3986.67
6       606  1840.14
> sinistres <- merge(somme,nombre,all=TRUE)
> sinistres$cout=replace(sinistres$cout,is.na(sinistres$cout),0)
```

On peut alors utiliser les modèles Tweedie, par exemple avec un paramètre p=1.3,

```
> regTw <- glm(cout~ageconducteur+agevehicule+as.factor(puissance),
+ data=sinistres, tweedie(1.3,0))
> summary(regTw)

Call:
glm(formula = cout ~ ageconducteur + agevehicule, family = tweedie(puis,
    0), data = sinistres)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
 -19.384    -8.516    -8.019    -7.481  2389.532

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.947441   0.224142  22.073  < 2e-16 ***
ageconducteur -0.014533   0.004525  -3.212  0.00132 **
agevehicule    0.023133   0.010198   2.268  0.02330 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for Tweedie family taken to be 59123.96)

    Null deviance: 106887434  on 678012  degrees of freedom
Residual deviance: 105804487  on 678010  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 8
```

Chapitre 2

Les provisions pour sinistres à payer

Dans ce chapitre, nous allons étudier les méthodes pour calculer le montant des *provisions pour sinistres à payer*, et plus particulièrement, des méthodes permettant de quantifier la marge d'erreur associée.

2.1 La problématique du provisionnment

Comme le définit [29], "*les provisions techniques sont les provisions destinées à permettre le règlement intégral des engagements pris envers les assurés et bénéficiaires de contrats. Elles sont liées à la technique même de l'assurance, et imposées par la règlementation*". D'un point de vue plus formel, à la date t, la compagnie d'assurance est tenue de constituer une provision pour les sinistres survenus avant la date t qu'elle sera tenu d'indemniser. Elle doit donc estimer le coût des sinistres survenus, et retrancher les montants déjà versés. Il s'agit donc fondamentalement d'un problème de prévision.

Parmi les méthodes reconnues par les autorités de contrôles, les plus classiques sont basées sur les cadences de paiements. On raisonne pour cela par année de survenance de sinistre, et on suppose une certaine régularité dans la cadence de paiement.

2.1.1 Quelques définitions et notations, aspects règlementaires et comptables

La plupart des méthodes présentées ici sont détaillées dans [7], ou [33]. Classiquement, on notera

- i (en ligne) l'année de survenance,
- j (en colonne) l'année de développement,
- $Y_{i,j}$ les *incréments de paiments*, pour l'année de développement j, pour les sinistres survenus l'année i, tableau 2.1

- $C_{i,j}$ les *paiments cumulés*, au sens où $C_{i,j} = Y_{i,0} + Y_{i,1} + \cdots + Y_{i,j}$, pour l'année de survenance j, tableau 2.2
- P_i la prime acquise pour l'année i, tableau 2.3
- $N_{i,j}$ le *nombre cumulé de sinistres* pour l'année de survenance i vu au bout de j années, tableau 2.4

	0	1	2	3	4	5
0	3209	1163	39	17	7	21
1	3367	1292	37	24	10	
2	3871	1474	53	22		
3	4239	1678	103			
4	4929	1865				
5	5217					

TABLE 2.1 – Triangle des incréments de paiements, $\boldsymbol{Y} = (Y_{i,j})$.

	0	1	2	3	4	5
0	3209	4372	4411	4428	4435	4456
1	3367	4659	4696	4720	4730	
2	3871	5345	5398	5420		
3	4239	5917	6020			
4	4929	6794				
5	5217					

TABLE 2.2 – Triangle des paiements cumulés, $\boldsymbol{C} = (C_{i,j})$.

Year i	0	1	2	3	4	5
P_i	4591	4672	4863	5175	5673	6431

TABLE 2.3 – Vecteur des primes acquises, $\boldsymbol{P} = (P_i)$.

Formellement, toutes ces données sont stockées dans des matrices, avec des valeurs manquantes `NA` pour les valeurs futures. Pour les importer, on utilisera les triangles `PAID`, `PREMIUM`, `NUMBER` et `INCURRED`

```
> source(bases.R)
> PAID
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 3209 4372 4411 4428 4435 4456
[2,] 3367 4659 4696 4720 4730   NA
[3,] 3871 5345 5398 5420   NA   NA
```

	0	1	2	3	4	5
0	1043.4	1045.5	1047.5	1047.7	1047.7	1047.7
1	1043.0	1027.1	1028.7	1028.9	1028.7	
2	965.1	967.9	967.8	970.1		
3	977.0	984.7	986.8			
4	1099.0	1118.5				
5	1076.3					

TABLE 2.4 – Triangle des nombres de sinistres, cumulés, en milliers, $\boldsymbol{N} = (N_{i,j})$.

```
[4,] 4239 5917 6020   NA   NA   NA
[5,] 4929 6794   NA   NA   NA   NA
[6,] 5217   NA   NA   NA   NA   NA
```

Le triangle des incréments se déduit facilement du triangle des cumulés

```
> INCREMENT <- PAID
> INCREMENT[,2:nrow(PAID)] <- PAID[,2:nrow(PAID)]-PAID[,1:(nrow(PAID)-1)]
> INCREMENT
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 3209 1163   39   17    7   21
[2,] 3367 1292   37   24   10   NA
[3,] 3871 1474   53   22   NA   NA
[4,] 4239 1678  103   NA   NA   NA
[5,] 4929 1865   NA   NA   NA   NA
[6,] 5217   NA   NA   NA   NA   NA
```

2.1.2 Formalisation du problème du provisionnement

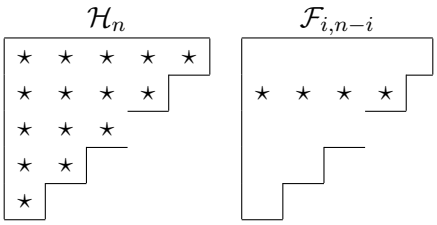
Le provisionnement est un problème de prédiction, conditionelle à l'information dont on dispose à la date n. On notera $\mathcal{F}_n$ l'information disponible à la date n, soit formellement

$$\mathcal{H}_n = \{(X_{i,j}), i+j \leq n\} = \{(C_{i,j}), i+j \leq n\}$$

On cherche à étudier, par année de survenance, la loi conditionnelle de $C_{i,\infty}$ sachant $\mathcal{H}_n$, ou encore, si l'on suppose les sinistres clos au bout de n années la loi de $C_{i,n}$ sachant $\mathcal{H}_n$. Si l'on se focalise sur une année de survenance particulière, on pourra noter

$$\mathcal{F}_{i,n-i} = \{(X_{i,j}), j = 0, \cdots, n-i)\} = \{(C_{i,j}), j = 0, \cdots, n-i)\}$$

Cette notation permet de prendre en compte que l'information disponible change d'une ligne à l'autre.



On cherchera par la suite à prédire le montant des sinistres à payer pour l'année i, i.e.

$$\widehat{C}_{i,n}^{(n-i)} = \mathbb{E}[C_{i,n}|\mathcal{F}_{i,n-i}]$$

et la différence entre ce montant et le montant déjà payé constituera la provision pour sinistres à payer,

$$\widehat{R}_i = \widehat{C}_{i,n}^{(n-i)} - C_{i,n-i}$$

On essayera ensuite de quantifier l'incertitude associée à cette prédiction. Comme on le verra les méthodes usuelles visaient à calculer

$$\mathrm{Var}[C_{i,n}|\mathcal{F}_{i,n-i}] \text{ ou } \mathrm{Var}[\widehat{C}_{i,n}^{(n-i)}]$$

ce que l'on appelera incertitude à horizon ultime. Mais ce n'est pas ce que propose Solvabilité II, demandant plutôt de mesurer une incertitude dite *à un an*. Pour cela, on va s'intéresser à la prédiction qui sera faite dans un an,

$$\widehat{C}_{i,n}^{(n-i+1)} = \mathbb{E}[C_{i,n}|\mathcal{F}_{i,n-i+1}]$$

et plus particulièrement le changement dans l'estimation de la charge ultime

$$\Delta_i^n = \widehat{C}_{i,n}^{(n-i+1)} - \widehat{C}_{i,n}^{(n-i)}.$$

Si cette différence est positive, on parle de *mali* (il faudra gonfler la provision afin de pouvoir payer les sinistres), et si elle est négative, on parle de *boni*. On peut montrer que

$$\mathbb{E}[\Delta_i^n|\mathcal{F}_{i,n-i}] = 0,$$

autrement dit, on ne peut espérer faire ni boni, ni mali, en moyenne. Les contraintes règlementaires imposéeés par Solvabilité II demandent de calculer

$$\mathrm{Var}[\Delta_i^n|\mathcal{F}_{i,n-i}].$$

2.2 Les cadences de paiements et la méthode Chain Ladder

L'utilisation des cadences de paiements pour estimer la charge future date du début du XXème siècle. On suppose qu'il existe une relation de récurrence de la forme

$$C_{i,j+1} = \lambda_j C_{i,j} \text{ pour tout } i, j = 1, \cdots, n.$$

Un estimateur naturel pour λ_j, basé sur l'expérience passée est alors

$$\widehat{\lambda}_j = \frac{\sum_{i=1}^{n-j} C_{i,j+1}}{\sum_{i=1}^{n-j} C_{i,j}} \text{ pour tout } j = 1, \cdots, n-1.$$

De telle sorte que l'on peut alors prédire la charge pour la partie non-observée dans le triangle,

$$\widehat{C}_{i,j} = \left[\widehat{\lambda}_{n+1-i} ... \widehat{\lambda}_{j-1}\right] C_{i,n+1-i}.$$

```
> n <- nrow(PAID)
> LAMBDA <- rep(NA,n-1)
> for(i in 1:(n-1)){
+  LAMBDA[i] <- sum(PAID[1:(n-i),i+1])/
+               sum(PAID[1:(n-i),i]) }
```

Notons qu'au lieu de calculer les facteurs de développement, on peut aussi des taux de développement, cumulés ou non. Autrement dit, au lieu d'écrire $C_{i,j+1} = \lambda_j C_{i,j}$ pour tout $i, j = 1, \cdots, n$, on suppose que

$$C_{i,j} = \gamma_j C_{i,n} \text{ ou } Y_{i,j} = \varphi_j C_{i,n}.$$

On notera que

$$\gamma_j = \prod_{k=j+1}^{n} \frac{1}{\lambda_k} \text{ et } \varphi_j = \begin{cases} \gamma_1 & \text{si } j = 1 \\ \gamma_j - \gamma_{j-1} & \text{si } j > 1 \end{cases}$$

```
>   cat("Lambda =",LAMBDA)
Lambda = 1.380933 1.011433 1.004343 1.001858 1.004735
> GAMMA <- rev(cumprod(rev(1/LAMBDA)))
>   cat("Gamma =", GAMMA)
Gamma = 0.7081910 0.9779643 0.9891449 0.9934411 0.9952873
>   cat("Phi =", c(GAMMA[1],diff(GAMMA)))
Phi = 0.708191033 0.269773306 0.011180591 0.004296183 0.001846141
```

	0	1	2	3	4	n
λ_j	1,38093	1,01143	1,00434	1,00186	1,00474	1,0000
γ_j	70,819%	97,796%	98,914%	99,344%	99,529%	100,000%
φ_j	70,819%	26,977%	1,118%	0,430%	0,185%	0,000%

TABLE 2.5 – Facteurs de développement, $\widehat{\boldsymbol{\lambda}} = (\widehat{\lambda}_i)$, exprimés en cadence de paiements par rapport à la charge utlime, en cumulé (i.e. $\widehat{\boldsymbol{\gamma}}$), puis en incréments (i.e. $\widehat{\boldsymbol{\varphi}}$).

On notera qu'il est possible de voir l'estimateur Chain-Ladder comme une moyenne pondérée des facteurs de transition individuels, i.e.

$$\widehat{\lambda}_j = \sum_{i=1}^{n-j} \omega_{i,j}\lambda_{i,j} \text{ où } \omega_{i,j} = \frac{C_{i,j}}{\sum_{i=1}^{n-j} C_{i,j}} \text{ et } \lambda_{i,j} = \frac{C_{i,j+1}}{C_{i,j}}.$$

Aussi, on peut obtenir ces coefficients à l'aide de régressions linéaires pondérées sans constantes, en régressant les $C_{\cdot,j+1}$ sur les $C_{\cdot,j}$. Ainsi, pour la première valeur,

```
> x <- PAID[,1]
> y <- PAID[,2]
> lm(y ~ x + 0, weights=1/x)
Call:
lm(formula = y ~ x + 0, weights = 1/x)
Coefficients:
x
1.381
```

Une fois estimé le facteur de développement, rien de plus simple que de compléter le triangle,

```
> TRIANGLE <- PAID
> for(i in 1:(n-1)){
+ TRIANGLE[(n-i+1):(n),i+1]=LAMBDA[i]*TRIANGLE[(n-i+1):(n),i]}
```

	0	1	2	3	4	5
0	3209	4372	4411	4428	4435	4456
1	3367	4659	4696	4720	4730	4752.4
2	3871	5345	5398	5420	5430.1	5455.8
3	4239	5917	6020	6046.15	6057.4	6086.1
4	4929	6794	6871.7	6901.5	6914.3	6947.1
5	5217	7204.3	7286.7	7318.3	7331.9	7366.7

TABLE 2.6 – Triangle des paiements cumulés, $\boldsymbol{C} = (C_{i,j})_{i+j\leq n}$ avec leur projection future $\widehat{\boldsymbol{C}} = (\widehat{C}_{i,j})_{i+j>n}$

Le montant de provisions est alors la différence entre ce que l'on pense payer pour chaque année de survenance (la dernière colonne) et que ce l'on a déjà payé (la seconde diagonale)

```
> ultimate <- TRIANGLE.D[,6]*(1+0.00)
> payment.as.at <- diag(TRIANGLE.D[,6:1])
> RESERVES <- ultimate-payment.as.at
> cat("Total reserve =",RESERVES)
Total reserve = 0.000 22.391 35.793 65.677 153.368 2149.656
```

On note qu'ici `sum(RESERVES)` vaut `2426.885`, ce qui correspond au montant total de réserves qu'il convient d'allouer.

Un algorithme plus rapide est d'utiliser directement la formule basée sur le produit des coefficients de transition. On a alors

```
> DIAG <- diag(triangle[,n:1])
> PRODUIT <- c(1,rev(LAMBDA))
> sum((cumprod(PRODUIT)-1)*DIAG))
> 2426.885
```

2.3 De Mack à Merz & Wüthrich

La méthode dite *Chain Ladder*, que nous venons de voir, est une méthode dite déterministe, au sens où l'on ne construit pas de modèle probabiliste permettant de mesurer l'incertitude associée à la prédiction du montant des réserves. Différents modèles ont été proposés à partir des années 90, à partir du modèles de Mack, jusqu'à l'approche proposée par Merz & Wüthrich qui introduira la notion d'*incertitude à un an*.

2.3.1 Quantifier l'incertitude dans une prédiction

Nous avons obtenu, par la méthode Chain Ladder un estimateur du montant de provision, $\widehat{R}$. Classiquement, pour quantifier l'erreur associée à un estimateur, on calcul la *mean squared error* - mse - associée,

$$\mathbb{E}([\widehat{R}-R]^2)$$

Formellement, comme R est ici une variable aléatoire, on ne parle pas de mse, mais de mse *de prédiction*, notée msep (on ne prédit pas sur les données passées, mais on utilisera les donnéees pour calibrer un modèle qui servira ensuite à faire de la prédiction pour les années futures). Aussi

$$\text{msep}(\widehat{R}) = \mathbb{E}([\widehat{R}-R]^2).$$

Ce terme peut se décomposer en deux (en faisant une approximation au premier ordre), au sens où

$$\mathbb{E}([\widehat{R}-R]^2) \approx \underbrace{\mathbb{E}([\widehat{R}-\mathbb{E}(R)]^2)}_{\text{mse}(\widehat{R})} + \underbrace{\mathbb{E}([R-\mathbb{E}(R)]^2)}_{\text{Var}(R)}$$

où le terme de gauche est l'erreur d'estimation, compte tenu du fait que nous avons dû estimer le montant de provisions à partir de la partie supérieure du triangle, et le terme de droite est l'erreur classique de modèle (tout modèle comportant une partie résiduelle orthogonale aux observations, et donc imprévisible).

En fait, en toute rigueur (et nous en aurons besoin par la suite), on cherche plutôt à calculer un msep conditionnel à l'information dont on dispose au bout de n années,

$$\text{msep}_n(\widehat{R}) = \mathbb{E}([\widehat{R} - R]^2|\mathcal{H}_n).$$

2.3.2 Le formalisme de Mack

[20] a proposé un cadre probabiliste afin de justifier l'utilisation de la méthode Chain-Ladder. Pour cela, on suppose que $(C_{i,j})_{j\geq 0}$ est un processus Markovien, et qu'il existe $\boldsymbol{\lambda} = (\lambda_j)$ et $\boldsymbol{\sigma} = (\sigma_j^2)$ tels que

$$\left\{ \begin{array}{l} \mathbb{E}(C_{i,j+1}|\mathcal{H}_{i+j}) = \mathbb{E}(C_{i,j+1}|C_{i,j}) = \lambda_j \cdot C_{i,j} \\ \text{Var}(C_{i,j+1}|\mathcal{H}_{i+j}) = \text{Var}(C_{i,j+1}|C_{i,j}) = \sigma_j^2 \cdot C_{i,j} \end{array} \right.$$

On note que sous ces hypothèses,

$$\mathbb{E}(C_{i,j+k}|\mathcal{H}_{i+j}) = \mathbb{E}(C_{i,j+k}|C_{i,j}) = \lambda_j \cdot \lambda_{j+1} \cdots \lambda_{j+k-1} C_{i,j}$$

[20] rajoute une hypothèse supplémentaire d'indépendance entre les années de survenance, autrement dit $(C_{i,j})_{j=1,\ldots,n}$ et $\left(C_{i',j}\right)_{j=1,\ldots,n}$ sont indépendant pour tout $i \neq i'$.

Une réécriture du modèle est alors de supposer que

$$C_{i,j+1} = \lambda_j C_{i,j} + \sigma_j \sqrt{C_{i,j}} + \varepsilon_{i,j}$$

où les résidus $(\varepsilon_{i,j})$ sont i.i.d. et centrés. A partir de cette écriture, il peut paraître légitime d'utiliser les méthodes des moindres carrés pondérés pour estimer ces coefficients, en notant que les poids doivent être inversement proportionnels à la variance, autrement dit aux $C_{i,j}$, i.e. à j donné, on cherche à résoudre

$$\min \left\{ \sum_{i=1}^{n-j} \frac{1}{C_{i,j}} \left(C_{i,j+1} - \lambda_j C_{i,j} \right)^2 \right\}$$

Pour tester ces deux premières hypothèses, on commence par représenter les $C_{\cdot,j+1}$ en fonction des $C_{\cdot,j}$ à j donné. Si la première hypothèse est vérifiée, les points doivent être alignés suivant une droite passant par l'origine.

```
> j=1
> plot(PAID[,j],PAID[,j+1],pch=19,col="red",cex=1.5)
> abline(lm(PAID[,j+1]~0+PAID[,j]),col="blue",lwd=2)
```

La Figure 2.1 montre ainsi les nuages de points pour `j=1` et `j=2`.

Pour la seconde, on peut étudier les résidus standardisés ([20] parle de *weighted residuals*), $\epsilon_{i,j} = \dfrac{C_{i,j+1} - \widehat{\lambda}_j C_{i,j}}{\sqrt{C_{i,j}}}$.

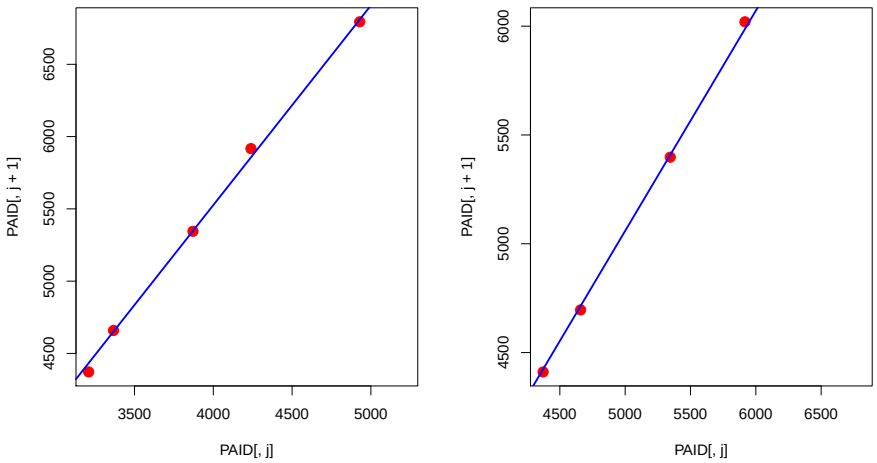


FIGURE 2.1 – Nuage de points $C_{\cdot,j+1}$ en fonction des $C_{\cdot,j}$ pour $j = 1, 2$, et droite de régression passant par l'origine.

```
> j=1
> RESIDUS <- (PAID[,j+1]-LAMBDA[j]*PAID[,j])/sqrt(PAID[,j])
```

L'utilisation des résidus standardisés nous donnent d'ailleurs une idée simple pour estimer le paramètre de volatilité.

$$\widehat{\sigma}_j^2 = \frac{1}{n-j-1} \sum_{i=0}^{n-j-1} \left(\frac{C_{i,j+1} - \widehat{\lambda}_j C_{i,j}}{\sqrt{C_{i,j}}} \right)^2$$

ce qui peut aussi s'écrire

$$\widehat{\sigma}_j^2 = \frac{1}{n-j-1} \sum_{i=0}^{n-j-1} \left(\frac{C_{i,j+1}}{C_{i,j}} - \widehat{\lambda}_j \right)^2 \cdot X_{i,j}$$

(ce qui est à rapprocher de l'écriture du facteur de transition λ comme moyenne pondérée des facteurs de transitions observés).

```
> SIGMA=sqrt(SIGMA)
> f <- PAID[,2:n]/PAID[,1:(n-1)]
> SIGMA <- rep(NA,n-1)
> for(i in 1:(n-1)){
+ D <- PAID[,i]*(f[,i]-t(rep(LAMBDA[i],n)))^2
+ SIGMA[i] <- 1/(n-i-1)*sum(D[,1:(n-i)])}
> SIGMA[n-1] <- min(SIGMA[(n-3):(n-2)])
```

```
> SIGMA=sqrt(SIGMA)
> cat("Volatilite =", SIGMA)
Volatilite = 0.7248578 0.3203642 0.04587297 0.02570564 0.02570564
```

Cette méthode permet d'estimer les différents paramètres intervenants dans le modèle de [20].

2.3.3 La notion de *tail factor*

Classiquement on suppose que la première ligne de notre triangle est close : il n'y a plus de sinistres ouverts, et donc le montant de provision pour cette année de survenance est nul. Cette ligne servira de base pour tous les développements ultérieurs. Cette hypothèse peut être un peu trop forte pour les branches à déroulement long. [21] a posé les bases des premiers modèles toujours utilisés. On supposera qu'il existe alors un $\lambda_\infty > 1$ tel que

$$C_{i,\infty} = C_{i,n} \times \lambda_\infty.$$

Une méthode qui a souvent été utilisée a reposé sur l'idée que l'on pouvait projeter les λ_i par une extrapolation exponentielle (ou une extrapolation linéaire des $\log(\lambda_k - 1)$), puis on pose

$$\lambda_\infty = \prod_{k \geq n} \widehat{\lambda}_k$$

Mais mieux vaut faire attention, en particulier s'il y a des valeurs aberrantes.

```
> logL <- log(LAMBDA-1)
> tps <- 1:5
> modeL <- lm(logL~tps)
> tpsP <- seq(6:1000)
> logLP <- predict(modeL,newdata=data.frame(tps=tpsP))
> lambda <- prod(exp(logLP)+1)
> lambda
> cat("Facteur moyen =", lambda)
Facteur moyen = 1.146162
```

Autrement dit, cette méthode prévoit de rajouter 14% de charge par rapport à la prédiction faite par les méthodes classiques, en supposant la première année close.

2.3.4 Des estimateurs des paramètres à l'incertitude sur le montant des provisions

A partir de tous ces estimateurs, on peut estimer le msep du montant de provision par année de survenance, $\widehat{R}_i$, mais aussi agrégé, toutes années de survenances confondues. Les formules sont données dans [21] ou [7].

On peut aussi utiliser la fonction `MackChainLadder` de `library(ChainLadder)`.

```
> library(ChainLadder)
> MackChainLadder(PAID)
MackChainLadder(Triangle = PAID)

  Latest Dev.To.Date Ultimate    IBNR Mack.S.E CV(IBNR)
1  4,456       1.000    4,456     0.0     0.000      NaN
2  4,730       0.995    4,752    22.4     0.639   0.0285
3  5,420       0.993    5,456    35.8     2.503   0.0699
4  6,020       0.989    6,086    66.1     5.046   0.0764
5  6,794       0.978    6,947   153.1    31.332   0.2047
6  5,217       0.708    7,367 2,149.7    68.449   0.0318

              Totals
Latest:     32,637.00
Ultimate:   35,063.99
IBNR:        2,426.99
Mack S.E.:      79.30
CV(IBNR):        0.03
```

On retrouve l'estimation du montant total de provisions $\widehat{R}$, `IBNR`, qui vaut `2,426.99`, ainsi que msep$(\widehat{R})$ correspondant au `Mack S.E.` qui vaut ici `79.30`. Les informations par année de survenance i sont indiqués dans la première partie du tableau.

On obtient également plusieurs graphiques en utilisant la fonction `plot()`, correspondant aux Figures 2.2, 2.3 et 2.4

2.3.5 Un mot sur Munich-Chain Ladder

La méthode dite *Munich-Chain-Ladder*, developpée dans [27], propose d'utiliser non seulement les paiements cumulés, mais aussi une autre information disponible : l'estimation des charges des différents sinistres faites par les gestionnaires de sinistres. Les triangles de paiements étaient basés sur des mouvements financiers ; ces triangles de charges sont basées sur des estimations faites par des gestionnaires compte tenu de l'information à leur disposition. Les sinistres tardifs ne sont pas dedans, et certains sinistres seront classés sans suite. Toutefois, il peut paraître légimite d'utiliser cette information.

Dans la méthode *Munich-Chain-Ladder*, on dispose des triangles $(C_{i,j})$ correspond aux incréments de paiments, et $(\Gamma_{i,j})$ aux charges dites dossier/dossier. En reprenant les notations de [27] on définie les ratio paie-

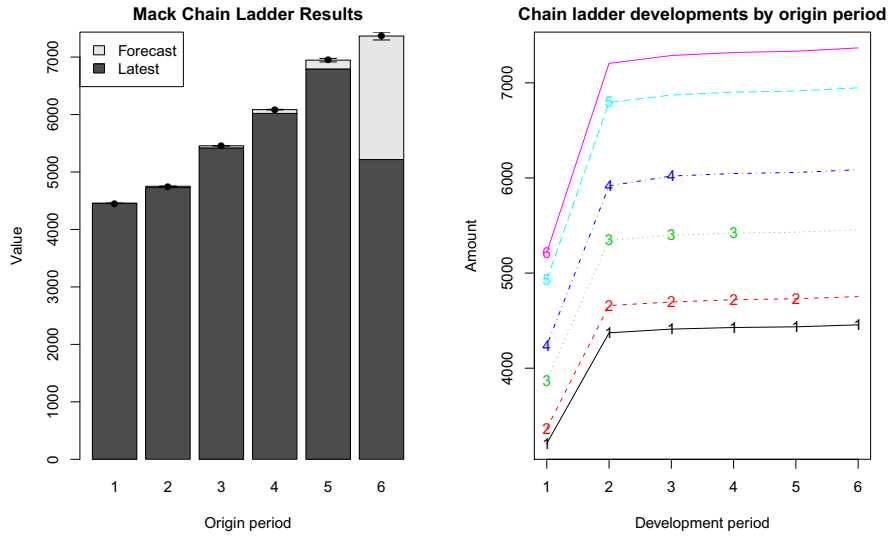


FIGURE 2.2 – Comparaison entre la charge finale estimée et la somme déjà payée, à gauche, et les cadences de paiements prédites par la méthode Chain Ladder.

ment/charge, et charge/paiement,

$$Q_{i,j} = \frac{C_{i,j}}{\Gamma_{i,j}} \text{ et } Q_{i,j}^{-1} = \frac{\Gamma_{i,j}}{C_{i,j}}$$

Comme dans le modèle Chain-Ladder de base, on suppose que

$$\begin{cases} \mathbb{E}(C_{i,j+1|\mathcal{F}^C_{i+j}}) = \lambda_j^C C_{i,j} \text{ et } \mathrm{Var}(C_{i,j+1|\mathcal{F}^C_{i+j}}) = [\sigma_j^C]^2 C_{i,j} \\ \mathbb{E}(\Gamma_{i,j+1|\mathcal{F}^\Gamma_{i+j}}) = \lambda_j^\Gamma \Gamma_{i,j} \text{ et } \mathrm{Var}(\Gamma_{i,j+1|\mathcal{F}^\Gamma_{i+j}}) = [\sigma_j^\Gamma]^2 \Gamma_{i,j} \end{cases}$$

On rajoute également une information sur les $\lambda_{i,j}$. Si

$$\lambda^C_{i,j-1} = \frac{C_{i,j}}{C_{i,j-1}} \text{ et } \lambda^\Gamma_{i,j-1} = \frac{\Gamma_{i,j}}{\Gamma_{i,j-1}}$$

on suppose que

$$\mathbb{E}(\lambda^C_{i,j-1}|\mathcal{F}_{i+j}) = \lambda^C_{j-1} + \lambda^C \sqrt{\mathrm{Var}(\lambda^C_{i,j-1}|\mathcal{F}^C_{i+j})} \cdot \frac{Q^{-1}_{i,j-1} - \mathbb{E}(Q^{-1}_{i,j-1}|\mathcal{F}_{i+j^C})}{\sqrt{\mathrm{Var}(Q^{-1}_{i,j-1}|\mathcal{F}_{i+j^C})}}$$

et

$$\mathbb{E}(\lambda^\Gamma_{i,j-1}|\mathcal{F}_{i+j}) = \lambda^\Gamma_{j-1} + \lambda^\Gamma \sqrt{\mathrm{Var}(\lambda^\Gamma_{i,j-1}|\mathcal{F}_{i+j})^\Gamma} \cdot \frac{Q_{i,j-1} - \mathbb{E}(Q_{i,j-1}|\mathcal{F}_{i+j^\Gamma})}{\sqrt{\mathrm{Var}(Q_{i,j-1}|\mathcal{F}_{i+j^\Gamma})}}$$

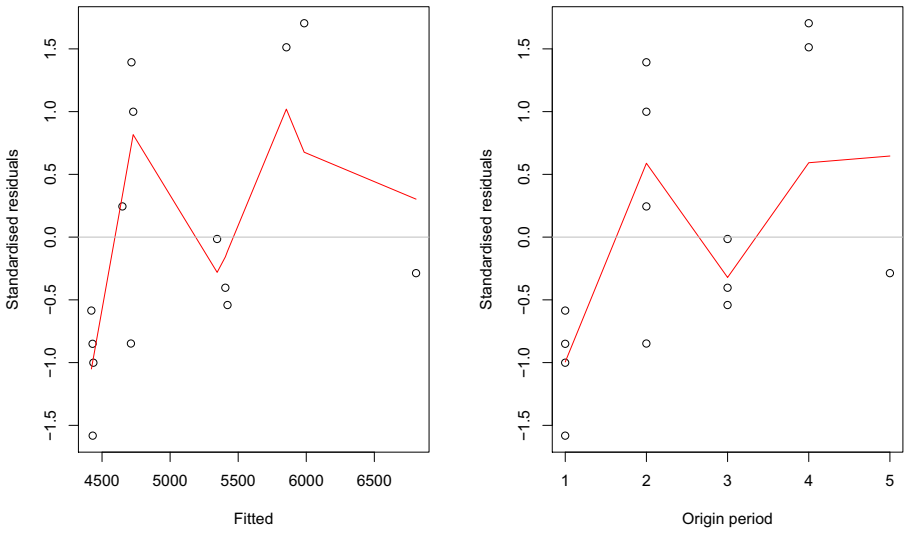


FIGURE 2.3 – Evolution des résidus standardisés en fonction des $\widehat{C}_{i,j}$ et des i.

On notera qu'il s'agit d'une extension du modèle Chain-Ladder, et en particulier

$$\mathbb{E}(\lambda^{\Gamma}_{i,j-1}|\mathcal{F}^{C}_{i+j}) = \mathbb{E}[\mathbb{E}(\lambda^{\Gamma}_{i,j-1}|\mathcal{F}_{i+j})|\mathcal{F}_{i+j})^{C}] = \lambda^{C}_{j-1}.$$

Les termes λ^{C} et λ^{Γ} sont alors simplement des coefficients de corrélation conditionnelle. Plus précisément

$$\lambda^{C} = \mathrm{Cor}(\Gamma_{i,j-1}, C_{i,j}|\mathcal{F}^{C}_{i+j-1})$$

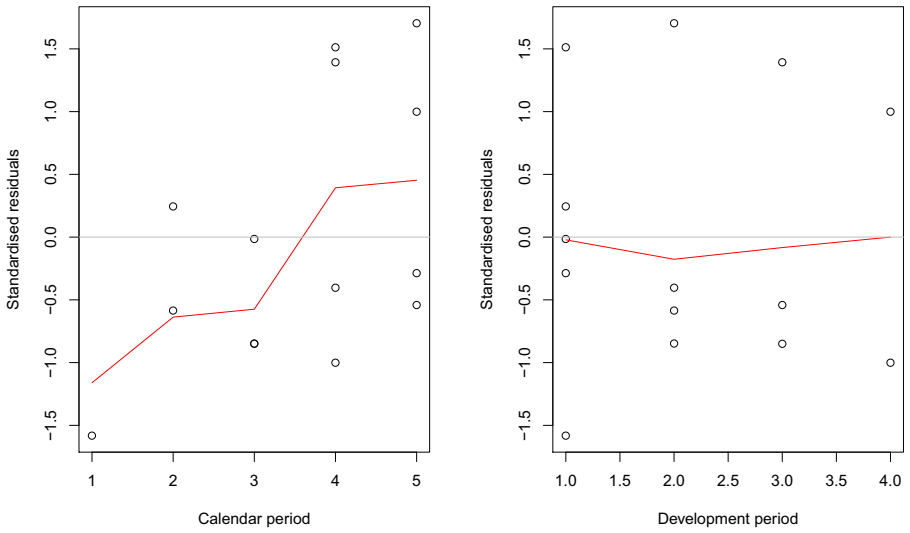
Sous ces hypothèses, il est possible de construire des estimateurs sans biais de $\mathbb{E}(C_{i,j}|C_{i,j-1})$, de $\mathbb{E}(\Gamma_{i,j}|\Gamma_{i,j-1})$, de $\mathbb{E}(Q_{i,j}|\mathcal{F}^{\Gamma}_{i+j})$ et de $\mathbb{E}(Q^{-1}_{i,j}|\mathcal{F}^{C}_{i+j})$.

Pour estimer les deux dernières quantités, posons

$$\widehat{Q}_j = \frac{\sum_{i=0}^{n_j} C_{i,j}}{\sum_{i=0}^{n_j} \Gamma_{i,j}} = \frac{1}{\widehat{Q^{-1}}_j}$$

On peut aussi estimer les variances conditionnelles. Par exemple

$$\widehat{\mathrm{Var}}(Q_{i,j}||\mathcal{F}^{\Gamma}_{i+j}) = ()^{-1} \sum_{i=0}^{n-j} \Gamma_{i,j}[Q_{i,j} - \widehat{Q}_j]^2$$



et une expression analogue pour $\widehat{\mathrm{Var}}(Q_{i,j}^{-1}|\mathcal{F}_{i+j}^{C})$.

A partir de ces quantités, posons enfin

$$\tilde{Q}_{i,j} = \frac{Q_{i,j} - \widehat{Q}_j}{\sqrt{\widehat{\mathrm{Var}}(Q_{i,j}|\mathcal{F}_{i+j}^{\Gamma})}},$$

$$\tilde{\lambda}_{i,j}^{\Gamma} = \frac{\sqrt{\Gamma_{i,j-1}}}{[\widehat{\sigma}_{j-1}^{I}]^2[\lambda_{i,j-1} - \widehat{\lambda}_{j-1}]}$$

et

$$\widehat{\lambda}^{\Gamma} = \frac{\sum \tilde{Q}_{i,j-1}\tilde{\lambda}_{i,j}^{\Gamma}}{\sum \tilde{Q}_{i,j-1}^2}$$

L'estimateur Munich-Chain-Ladder est construit de manière intérative. Le détails des formules est donné dans [27] ou [33].

```
> (MNCL=MunichChainLadder(Paid=PAID,
+ Incurred=INCURRED))
MunichChainLadder(Paid = PAID, Incurred = INCURRED)

  Latest Paid Latest Incurred Latest P/I Ratio Ult. Paid Ult. Incurred Ult. P/I R
1       4,456           4,456            1.000     4,456         4,456
2       4,730           4,750            0.996     4,753         4,750
```

	0	1	2	3	4	5
0	4795	4629	4497	4470	4456	4456
1	5135	4949	4783	4760	4750	
2	5681	5631	5492	5470		
3	6272	6198	6131			
4	7326	7087				
5	7353					

TABLE 2.7 – Triangle des estimations de charges dossier/dossier cumulées, $\mathbf{\Gamma} = (\Gamma_{i,j})$

```
3       5,420           5,470           0.991     5,455           5,454           1
4       6,020           6,131           0.982     6,086           6,085           1
5       6,794           7,087           0.959     6,983           6,980           1
6       5,217           7,353           0.710     7,538           7,533           1

Totals
            Paid Incurred P/I Ratio
Latest:   32,637   35,247      0.93
Ultimate: 35,271   35,259      1.00
```

De même que pour la fonction `MackChainLadder`, plusieurs graphiques peuvent être obtenus afin de mieux comprendre les évolutions des paiements, mais aussi de la charge dossier/dossier estimée par les gestionnaires de sinistres, présentés sur les Figures 2.5 et 2.6.

Si on compare les deux triangles, qui ont été complétés en tenant compte des interactions, on obtient des choses relativement proches,

```
> MNCL$MCLPaid
     [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 3209 4372.000 4411.000 4428.000 4435.000 4456.000
[2,] 3367 4659.000 4696.000 4720.000 4730.000 4752.569
[3,] 3871 5345.000 5398.000 5420.000 5429.716 5455.324
[4,] 4239 5917.000 6020.000 6046.090 6057.284 6085.875
[5,] 4929 6794.000 6890.045 6932.247 6949.447 6982.539
[6,] 5217 7251.382 7419.621 7478.759 7502.149 7538.194
> sum(MNCL$MCLPaid[,6]-diag(MNCL$MCLPaid[,6:1]))
[1] 2633.502
> MNCL$MCLIncurred
     [,1]     [,2]    [,3]     [,4]     [,5]     [,6]
[1,] 4975 4629.000 4497.00 4470.000 4456.000 4456.000
[2,] 5135 4949.000 4783.00 4760.000 4750.000 4750.415
[3,] 5681 5631.000 5492.00 5470.000 5454.691 5454.445
```

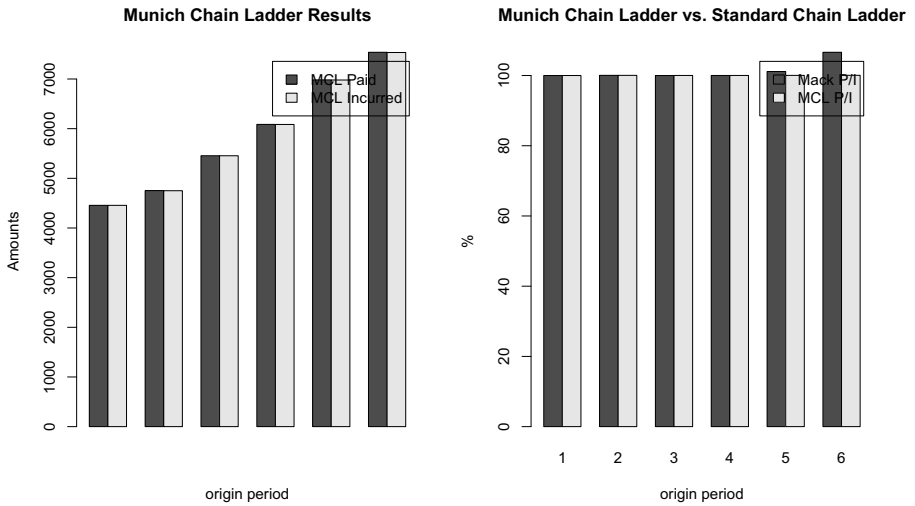


FIGURE 2.5 – Comparaison des méthodes Chain Ladder, et Munich Chain Ladder, en montant à gauche, et en valeurs relatives à droite.

```
[4,] 6272 6198.000 6131.00 6100.978 6084.986 6084.770
[5,] 7326 7087.000 6988.37 6984.274 6979.284 6979.732
[6,] 7353 7349.795 7493.64 7522.809 7532.206 7533.461
> sum(MNCL$MCLIncurred[,6]-diag(MNCL$MCLPaid[,6:1]))
[1] 2621.823
```

2.3.6 L'incertitude à un an de Merz & Wüthrich

[24] ont étudier la variation du boni/mali d'une année sur l'autre, c'est à dire du changement dans la prédiction de la charge totale. Ils ont en particulier montré que

$$\widehat{\text{msepc}}_{n-1}(\text{CDR}_i(t)) = \widehat{C}_{i,\infty}^2 \left(\widehat{\Gamma}_{i,n} + \widehat{\Delta}_{i,n}\right)$$

où

$$\widehat{\Delta}_{i,n} = \frac{\widehat{\sigma}_{n-i+1}^2}{\widehat{\lambda}_{n-i+1}^2 S_{n-i+1}^{n+1}} + \sum_{j=n-i+2}^{n-1} \left(\frac{C_{n-j+1,j}}{S_j^{n+1}}\right)^2 \frac{\widehat{\sigma}_j^2}{\widehat{\lambda}_j^2 S_j^n}$$

et

$$\widehat{\Gamma}_{i,n} = \left(1 + \frac{\widehat{\sigma}_{n-i+1}^2}{\widehat{\lambda}_{n-i+1}^2 C_{i,n-i+1}}\right) \prod_{j=n-i+2}^{n-1} \left(1 + \frac{\widehat{\sigma}_j^2}{\widehat{\lambda}_j^2 [S_j^{n+1}]^2} C_{n-j+1,j}\right) - 1$$

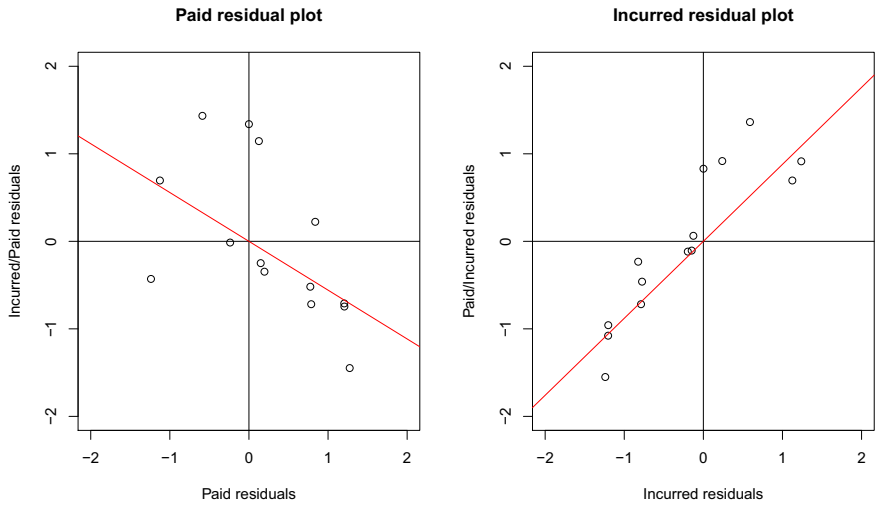


FIGURE 2.6 – Corrélations entre les triangles de développement des paiements, et des charges dossier/dossier.

[24] ont alors approché ce terme par

$$\widehat{\Gamma}_{i,n} \approx \frac{\widehat{\sigma}^2_{n-i+1}}{\widehat{\lambda}^2_{n-i+1} C_{i,n-i+1}} + \sum_{j=n-i+2}^{n-1} \left(\frac{C_{n-j+1,j}}{S^{n+1}_j} \right)^2 \frac{\widehat{\sigma}^2_j}{\widehat{\lambda}^2_j C_{n-j+1,j}}$$

en faisant tout simplement un développement de la forme $\prod(1+u_i) \approx 1 + \sum u_i$, mais qui n'est valide que si u_i est petit, soit ici

$$\frac{\widehat{\sigma}^2_j}{\widehat{\lambda}^2_j} << C_{n-j+1,j}.$$

Ces fonctions peuvent être obtenues à l'aide de la fonction `MSEP_Mack_MW` obtenue à l'aide de `source(MackMerzWuthrich.R)`. Pour expliquer rapidement les principales fonctions utilisées, il faut commencer par finir les matrices `Gamma` et `Delta`. Mais avant tout, initialement les matrices et vecteurs qui nous serviront par la suite,

```
I <- nrow(PAID)
J <- ncol(PAID)
diag <- diag_inv <- Phi <- Psi <- Delta <- Lambda <- Epsilon <- matrix(NA,I,1)
S_I <- S_II <- matrix(NA,1,J)
```

```
cov_obs <- cov_reel <- matrix(0,I,J)
msep_obs <- msep_reel <- matrix(NA,1,J+1)
Delta[1] <- Phi[1] <- Psi[1] <- Epsilon[1] <- Lambda[1] <- 0
```

Le plus simple est alors d'utiliser les sorties de `MackChainLadder`,

```
> Mack<-MackChainLadder(PAID)
> Mack$sigma[J-1] <- sqrt(min(Mack$sigma[J-2]^4/Mack$sigma[J-3]^2,
+                     min(Mack$sigma[J-2]^2,Mack$sigma[J-3]^2)))
> for (i in 1:I){
+        diag[i] = PAID[i,I-i+1]
+        diag_inv[i] = PAID[I-i+1,i]
+ }
> for (j in 1:J){
+        S_I[j] <- sum(PAID[1:(I-j),j])
+        S_II[j] <- sum(PAID[1:(I-j+1),j])
+ }
> S_I[I] <- 0
> Delta[2] <- Mack$sigma[I-1]^2/(S_I[I-1]*(Mack$f[I-1])^2)
> Phi[2] <- 0
> Psi[2] <- Mack$sigma[I-1]^2/(diag[2]*(Mack$f[I-1])^2)
> Epsilon[2] <- Mack$sigma[I-1]^2/(S_II[I-1]*(Mack$f[I-1])^2)
> Lambda[2] <- diag[2]*Mack$sigma[I-1]^2/((Mack$f[I-1]^2)*
+ S_II[I-1]*S_I[I-1])
```

pour les première valeurs. On fait ensuite une boucle pour incrémenter les vecteurs,

```
> for (i in 3:I){
+        Delta[i] <- Mack$sigma[I-i+1]^2/(S_I[I-i+1]*(Mack$f[I-i+1])^2) + sum(
+                     (diag_inv[(I-i+2):(J-1)]/S_II[(I-i+2):(J-1)])^2*
+                           Mack$sigma[(I-i+2):(J-1)]^2/(S_I[(I-i+2):(J-1)]*
+                           (Mack$f[(I-i+2):(J-1)])^2))
+        Phi[i] <- sum((diag_inv[(I-i+2):(J-1)]/S_II[(I-i+2):(J-1)])^2*
+                  Mack$sigma[(I-i+2):(J-1)]^2/(diag_inv[(I-i+2):(J-1)]*
+                  (Mack$f[(I-i+2):(J-1)])^2))
+        Psi[i] <- Mack$sigma[I-i+1]^2/(diag[i]*(Mack$f[I-i+1])^2)
+        Epsilon[i] <- Phi[i] + Mack$sigma[I-i+1]^2/(S_II[I-i+1]*
+        (Mack$f[I-i+1])^2)
+
+ Lambda[i] <- diag[i]*Mack$sigma[I-i+1]^2/((Mack$f[I-i+1]^2)*
+              S_II[I-i+1]*S_I[I-i+1]) + sum(
+                           (diag_inv[(I-i+2):(J-1)]/S_II[(I-i+2):(J-1)])^2*
+                           Mack$sigma[(I-i+2):(J-1)]^2/(S_I[(I-i+2):(J-1)]*
+                           (Mack$f[(I-i+2):(J-1)])^2))
```

```
+ }
> Gamma = Phi + Psi
```

Une fois ces matrices définies, on peut calculer les mse de prédiciton par année de survenance

```
for (i in 1:I){
        msep_obs[i] = Mack$FullTriangle[i,J]^2 * (Gamma[i] + Delta[i])
        msep_reel[i] = Mack$FullTriangle[i,J]^2 * (Phi[i] + Delta[i])
}
```

Pour l'incertitude associée au montant total de provision (toutes années confondues), il faut rajouter quelques calculs de covariance,

```
for (i in 2:(I-1)){
                for (k in (i+1):I){
                cov_obs[i,k] <- Mack$FullTriangle[i,J]*Mack$FullTriangle[k,J]*(Upsilon[i]
                cov_reel[i,k] <- Mack$FullTriangle[i,J]*Mack$FullTriangle[k,J]*(Phi[i] +
        }
}
```

On en déduit alors le mse de prédiction pour $\widehat{R}$,

```
> msep_obs[I+1] = sum(msep_obs[2:I]) + 2*sum(cov_obs)
> msep_reel[I+1] = sum(msep_reel[2:I]) + 2*sum(cov_reel)
```

Cette méthode correspond à l'approximation proposée par [24]. Pour faire le calcul exact, quelques petits changements sont à apporter (mais assez mineurs)

```
> facteur <- matrix(1,I,1)
> Phi_exact[2] <- Psi_exact[2] <- 0
> Gamma_exact[2] <- Mack$sigma[I-1]^2/(diag[I-1]*(Mack$f[I-1])^2)
> Epsilon_exact[2] <- Mack$sigma[I-1]^2/(S_II[I-1]*(Mack$f[I-1])^2)
> for (i in 3:I){
+        facteur[i] <-prod(1 + diag_inv[(I-i+2):(J-1)]*(Mack$sigma[(I-i+2):(J-1)]^2)/
+        ((S_II[(I-i+2):(J-1)]*Mack$f[(I-i+2):(J-1)])^2))
+        Phi_exact[i] <- (1 + Mack$sigma[I-i+1]^2/(diag_inv[I-i+1]*(Mack$f[I-i+1])^2))*
+                 (facteur[i]-1)
+        Psi_exact[i] <- (1 + Mack$sigma[I-i+1]^2/(S_II[i]*(Mack$f[I-i+1])^2)) * Phi[i] /
+                        (1 + Mack$sigma[I-i+1]^2/(diag_inv[I-i+1]*(Mack$f[I-i+1])^2))
+        Epsilon_exact[i] <- (1 + Mack$sigma[I-i+1]^2/(S_II[I-i+1]*(Mack$f[I-i+1])^2))*
+                            facteur[i]-1
+        Gamma_exact[i] <- (1 + Mack$sigma[I-i+1]^2/(diag_inv[I-i+1]*(Mack$f[I-i+1])^2))*
+                    facteur[i]-1
+}
```

ce qui permet d'obtenir la *vraie* valeur de $\widehat{\Gamma}_{i,n}$, stochée dans le vecteur `Gamma_exact`. A partir de là, on peut calculer les mse de prédiction, comme précédemment,

```
> for (i in 1:I){
+        msep_obs_exact[i] = Mack$FullTriangle[i,J]^2 * (Gamma_exact[i] + Delta[i
+        msep_reel_exact[i] = Mack$FullTriangle[i,J]^2 * (Phi_exact[i] + Delta[i]
+}
> for (i in 2:(I-1)){
+        for (k in (i+1):I){
+                cov_obs_exact[i,k] <- Mack$FullTriangle[i,J]*Mack$FullTriangle[k
+   (Upsilon_exact[i] + Lambda[i])
+                cov_reel_exact[i,k] <- Mack$FullTriangle[i,J]*Mack$FullTriangle
+                (Psi_exact[i] + Lambda[i])
+ }}
> msep_obs_exact[I+1] <- sum(msep_obs_exact[2:I]) + 2*sum(cov_obs_exact)
> msep_reel_exact[I+1] <- sum(msep_reel_exact[2:I]) + 2*sum(cov_reel_exact)
> msep_Mack <- array(0,c(1,I+1))
> msep_Mack[1:I] <- Mack$Mack.S.E[,I]
> msep_Mack[I+1] <- Mack$Total.Mack.S.E
> Vari <- array(0,c(1,I+1))
> for (i in 1:I){
+        Vari[i] <- Mack$FullTriangle[i,J]^2 * Psi[i]
+ }
> Vari[I+1] <- sum(Vari[1:I])
```

On dispose alors de l'ensemble des éléments permettant d'avoir une vision à un an de l'incertitude,

```
>        result <- cbind(t(msep_Mack), t(sqrt(msep_obs)), t(sqrt(msep_obs_exact))
>        result <- as.data.frame(result)
>        names(result) <- c("MSEP Mack","MSEP MW app.","MSEP MW ex.")
> result
   MSEP Mack MSEP MW app. MSEP MW ex.
1  0.0000000     0.000000    0.000000
2  0.6393379     1.424131    1.315292
3  2.5025153     2.543508    2.543508
4  5.0459004     4.476698    4.476698
5 31.3319292    30.915407   30.915407
6 68.4489667    60.832875   60.832898
7 79.2954414    72.574735   72.572700
```

2.4 Régression Poissonnienne et approches économétriques

Dans cette section, nous nous éloignerons des modèles récursifs inspirés de la méthode Chain Ladder, et nous reviendrons sur des classes de modèles très utilisés dans les années 70, appelés *modèles à facteurs*, remis au goût du jour en proposant une lecture économétrique de ces modèles, permettant ainsi d'obtenir des intervalles de confiance des différentes grandeurs.

2.4.1 Les modèles à facteurs, un introduction historique

Avant de présenter l'utilisation des modèles de régression, on peut commencer par évoquer des modèles plus anciens. Par exemple Taylor (1977) supposait que

$$Y_{i,j} = r_j \cdot \mu_{i+j}, \text{ pour tout } i, j$$

i.e. un effet colonne, de cadence de paiement, et un effet diagonal, que Taylor interprète comme un facteur d'inflation. Ce modèle peut se réécrire, dès lors qu'il n'y a pas d'incrément positif,

$$\log Y_{i,j} = \alpha_i + \gamma_{i+j}$$

qui prend alors une forme linéaire. On montrera par la suite que le cas

$$\log Y_{i,j} = \alpha_i + \beta_j$$

s'apparent à un modèle de type Chain-Ladder. En effet, cela suppose que

$$Y_{i,j} = a_i \times b_j$$

que l'on peut rapprocher du modèle de développement $Y_{i,j} = C_{i,n} \times \varphi_j$. [34] avait également proposé d'utiliser une courbe d'Hoerl, c'est à dire

$$\log Y_{i,j} = \alpha_i + \beta_i \cdot \log(j) + \gamma_i \cdot j.$$

2.4.2 Les modèles de de Vylder et de Chritophides

Les équations normales s'écrivent ici

$$\widehat{\alpha}_i = \frac{\sum_j Y_{i,j}\widehat{\beta}_j}{\sum_j \widehat{\beta}_j^2} \text{ et } \widehat{\beta}_j = \frac{\sum_i Y_{i,j}\widehat{\alpha}_i}{\sum_i \widehat{\alpha}_i^2},$$

ce qui ne résoud pas explicitement. Pour le résoudre, [4] a suggéré de le réécrire comme un modèle log-linéaire, i.e.

$$\log Y_{i,j} \sim \mathcal{N}(a_i + b_j, \sigma^2), \text{ pour tout } i, j$$

```
> an <- 6; ligne <- rep(1:an, each=an); colonne <- rep(1:an, an)
> INC <- PAID
> INC[,2:6] <- PAID[,2:6]-PAID[,1:5]
> Y <- as.vector(INC)
> lig  <- as.factor(ligne)
> col <- as.factor(colonne)
> reg <- lm(log(Y)~col+lig)
> summary(reg)

Call:
lm(formula = log(Y) ~ col + lig)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.9471     0.1101  72.188 6.35e-15 ***
col2          0.1604     0.1109   1.447  0.17849
col3          0.2718     0.1208   2.250  0.04819 *
col4          0.5904     0.1342   4.399  0.00134 **
col5          0.5535     0.1562   3.543  0.00533 **
col6          0.6126     0.2070   2.959  0.01431 *
lig2         -0.9674     0.1109  -8.726 5.46e-06 ***
lig3         -4.2329     0.1208 -35.038 8.50e-12 ***
lig4         -5.0571     0.1342 -37.684 4.13e-12 ***
lig5         -5.9031     0.1562 -37.783 4.02e-12 ***
lig6         -4.9026     0.2070 -23.685 4.08e-10 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1753 on 10 degrees of freedom
  (15 observations deleted due to missingness)
Multiple R-squared: 0.9975,     Adjusted R-squared: 0.9949
F-statistic: 391.7 on 10 and 10 DF,  p-value: 1.338e-11
```

On peut alors simplement utiliser cette régression pour construire le triangle de base du modèle, $\widehat{Y}_{i,j} = \exp[\widehat{a}_i + \widehat{b}_j]$ (la partie inférieure droite constituant la prédiction). Comme nous l'avions noté dans la Section 1.4.1, cet estimateur est toutefois biaisé,

```
>  logY <- predict(reg,newdata=data.frame(lig,col))
>  INCpred <- matrix(exp(logY),an,an)
>  INCpred
         [,1]     [,2]     [,3]     [,4]      [,5]     [,6]
[1,] 2827.436 1074.641 41.02692 17.99380  7.721706 21.00000
[2,] 3319.436 1261.638 48.16599 21.12488  9.065354 24.65419
```

```
[3,] 3710.527 1410.282 53.84084 23.61379 10.133422 27.55892
[4,] 5102.488 1939.333 74.03860 32.47222 13.934856 37.89732
[5,] 4917.944 1869.193 71.36081 31.29779 13.430869 36.52667
[6,] 5217.000 1982.857 75.70020 33.20098 14.247588 38.74783
```

Le montant de provision prédit est ici

```
>  cat("Total reserve =",sum(exp(logY[is.na(Y)==TRUE])))
Total reserve = 2444.02
```

ce qui est un légèrement différent de la prédiction obtenue par la méthode Chain Ladder. Si l'on corrige du biais (car $\exp(\mathbb{E}(\log(Y))) \neq \mathbb{E}(Y)$), on obtient alors

```
> sigma=summary(reg)$sigma
> INCpred <- matrix(exp(logY+sigma^2/2),an,an)
> INCpred
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 2871.209 1091.278 41.66208 18.27237  7.84125 21.32511
[2,] 3370.826 1281.170 48.91167 21.45193  9.20570 25.03588
[3,] 3767.972 1432.116 54.67438 23.97937 10.29030 27.98557
[4,] 5181.482 1969.357 75.18483 32.97495 14.15059 38.48403
[5,] 4994.082 1898.131 72.46559 31.78233 13.63880 37.09216
[6,] 5297.767 2013.554 76.87216 33.71498 14.46816 39.34771
> cat("Total reserve =",sum(exp(logY[is.na(Y)==TRUE]+sigma^2/2)))
Total reserve = 2481.857
```

2.4.3 La régression poissonnienne de Hachemeister & Stanard

[12], [18] et enfin [19] ont montré que dans une régression log-Poisson sur les incréments, la somme des prédictions des paiments à venir correspond à l'estimateur Chain Ladder. On retrouve ici un résultat pouvant être relié à la méthode des marges présentée dans la section 1.3.2.

```
> library(statmod)
> an <- 6; ligne <- rep(1:an, each=an); colonne <- rep(1:an, an)
> passe <- (ligne + colonne - 1)<=an; np <- sum(passe)
> futur <- (ligne + colonne - 1)> an; nf <- sum(passe)
> INC <- PAID
> INC[,2:6] <- PAID[,2:6]-PAID[,1:5]
> Y <- as.vector(INC)
> lig  <- as.factor(ligne)
> col <- as.factor(colonne)
> CL <- glm(Y~lig+col, family=quasipoisson)
```

```
> summary(CL)

Call:
glm(formula = Y ~ lig + col, family = quasipoisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3426  -0.4996   0.0000   0.2770   3.9355

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.05697    0.02769 290.995  < 2e-16 ***
lig2        -0.96513    0.02427 -39.772 2.41e-12 ***
lig3        -4.14853    0.11805 -35.142 8.26e-12 ***
lig4        -5.10499    0.22548 -22.641 6.36e-10 ***
lig5        -5.94962    0.43338 -13.728 8.17e-08 ***
lig6        -5.01244    0.39050 -12.836 1.55e-07 ***
col2         0.06440    0.03731   1.726 0.115054
col3         0.20242    0.03615   5.599 0.000228 ***
col4         0.31175    0.03535   8.820 4.96e-06 ***
col5         0.44407    0.03451  12.869 1.51e-07 ***
col6         0.50271    0.03711  13.546 9.28e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for quasipoisson family taken to be 3.18623)

    Null deviance: 46695.269  on 20  degrees of freedom
Residual deviance:    30.214  on 10  degrees of freedom
  (15 observations deleted due to missingness)
AIC: NA

Number of Fisher Scoring iterations: 4
```

Il y a ici un $2n-1$ paramètres à estimer, γ, $=(c_1,\cdots,c_{n-1})$ et $=(r_1,\cdots,r_{n-1})$. Compte tenu du choix des facteurs (ici un facteur ligne r et un facteur colonne c), une fois estimés ces paramètres, il est possible de *prédire* la partie inférieure du triangle très simplement, i.e.

$$\widehat{Y}_{i,j}=\widehat{\mu}_{i,j}=\exp[\widehat{\gamma}+\widehat{r}_i+\widehat{c}_j]$$

```
> mu.hat1 <- exp(predict(CL,newdata=data.frame(lig,col)))*futur
> cat("Total reserve =", sum(mu.hat1))
Total reserve = 2426.985
> mu.hat2 = predict(CL,newdata=data.frame(lig,col),type="response")*futur
```

```
> cat("Total reserve =", sum(mu.hat2))
Total reserve = 2426.985
```

On retrouve ici l'estimateur obtenu par la méthode Chain-Ladder.

La valeur de référence est la valeur dans le coin supérieur gauche. Compte tenu de la forme logarithmique du modèle, on a une interprétation simple de toutes les valeurs, relativement à cette première valeur

$$\mathbb{E}(Y_{i,j}|\mathcal{F}_n) = \mathbb{E}(Y_{0,0}|\mathcal{F}_n) \cdot \exp[r_i + c_j].$$

Remarque 2.1 *Dans certains triangle, il n'est pas rare d'avoir des* incréments négatifs.

2.4.4 Incertitude dans un modèle de régression

Nous avions noté auparavant qu'obtenir une estimation du montant de sinistres restant à payer ne suffisait pas, et qu'il fallait avoir un intervalle de confiance, au au moins une mesure de la dispersion du vrai montant autour de cette valeur prédite.

Les formules économétriques fermées

Les modèles de régressions pourraient paraître très intéressants car il existe des formules fermés pour toutes sortes de prédiction. Par exemple, dans une régression GLM avec un lien logarithmique, rappelons que

$$\mathbb{E}(Y_{i,j}|\mathcal{F}_n) = \mu_{i,j} = \exp[\eta_{i,j}]$$

ou encore

$$\widehat{Y}_{i,j} = \widehat{\mu}_{i,j} = \exp[\widehat{\eta}_{i,j}].$$

La *delta method* nous permet d'écrire que

$$\mathrm{Var}(\widehat{Y}_{i,j}) \approx \left|\frac{\partial \mu_{i,j}}{\partial \eta_{i,j}}\right|^2 \cdot \mathrm{Var}(\widehat{\eta}_{i,j}),$$

ce qui se simplifie dans le cas où le lien est logarithmique, i.e.

$$\frac{\partial \mu_{i,j}}{\partial \eta_{i,j}} = \mu_{i,j}$$

Aussi, pour une loi de Poisson surdispersée (comme dans [28]),

$$\mathbb{E}\left([Y_{i,j} - \widehat{Y}_{i,j}]^2\right) \approx \widehat{\phi} \cdot \widehat{\mu}_{i,j} + \widehat{\mu}_{i,j}^2 \cdot \widehat{\mathrm{Var}}(\widehat{\eta}_{i,j})$$

pour la partie inférieure du triangle. De plus, car il sera nécessaire de sommer tous les termes de la partie inférieure du triangle pour déterminer le montant total de provisions,

$$\mathrm{Cov}(\widehat{Y}_{i,j}, \widehat{Y}_{k,l}) \approx \widehat{\mu}_{i,j} \cdot \widehat{\mu}_{k,l} \cdot \widehat{\mathrm{Cov}}(\widehat{\eta}_{i,j}, \widehat{\eta}_{k,l}).$$

Le montant de provision que lon cherche à estimer étant la somme des prédictions de paiements à venir, $\widehat{R} = \sum_{i+j>n} \widehat{Y}_{i,j}$, alors

$$\mathbb{E}\left([R-\widehat{R}]^2\right) \approx \left(\sum_{i+j>n} \widehat{\phi}\cdot\widehat{\mu}_{i,j}\right) + \widehat{\boldsymbol{\mu}}'\cdot\widehat{\mathrm{Var}}(\widehat{\boldsymbol{\eta}})\cdot\widehat{\boldsymbol{\mu}}$$

Remarque 2.2 *Cette formule est malheureusement asymptotique, ce qui est rarement le cas en provisionnement où l'on dispose de très peu de données.*

Pour programmer cette formule, il faut écarter (un peu artificiellement) les valeurs manquantes.

```
>  Y2 <- Y; Y2[is.na(Y)] <- .001
>  CL2 <- glm(Y2~lig+col, family=quasipoisson)
>  p <- 2*an-1;
>  phi.P <- sum(residuals(CL,"pearson")^2)/(np-p)
>  Sig <- vcov(CL)
>  X <- model.matrix(CL2)
>  Cov.eta <- X%*%Sig%*%t(X)
>  mu.hat <- exp(predict(CL,newdata=data.frame(lig,col)))*futur
>  pe2 <- phi.P * sum(mu.hat) + t(mu.hat) %*% Cov.eta %*% mu.hat
>  cat("Total reserve =", sum(mu.hat), "mse =", sqrt(pe2),"\n")
Total reserve = 2426.985 mse = 131.7726
```

Les méthodes de simulations

Les méthodes de simulation sont une bonne alternative si on dispose de trop peu de données pour invoquer des théorèmes asymptotiques. Rappelons, comme le notait [20] qu'il existe 2 sources d'incertitude,

– l'erreur de modèle (on parle de *process error*)

– l'erreur d'estimation (on parle de *variance error*)

Il sera alors nécessaire d'utiliser deux algorithmes pour quantifier ces deux erreurs.

Afin de quatifier l'erreur d'estimation, il est naturel de simuler des faux triangles (supérieurs), puis de regarder la distribution des estimateurs de montant de provisions obtenus pour chaque triangles (par exemple par la méthode Chain Ladder, à l'aide de la fonction `chainladder` développée auparavant). A l'étape b, on génère un pseudo triangle à l'aide des résidus de Pearson. Rappelons que pour une régression de Poisson,

$$\widehat{\varepsilon}_{i,j} = \frac{Y_{i,j}-\widehat{m}_{i,j}}{\sqrt{\widehat{m}_{i,j}}}.$$

En simulant des erreurs (qui sont supposées indépendantes et identiquement distribuée), $\tilde{\boldsymbol{\varepsilon}}^b = (\tilde{\varepsilon}^b_{i,j})$, on pose alors

$$Y^b_{i,j} = \widehat{m}_{i,j} + \sqrt{\widehat{m}_{i,j}}\cdot\tilde{\varepsilon}^b_{i,j}.$$

Pour générer des erreurs, la méthode la plus usuelle est d'utiliser une simulation nonparamétrique, c'est à dire que l'on va bootstrapper les résidus parmi les pseudorésidus obtenus. Sinon il est aussi possible d'utiliser un modèle paramétrique (par exemple supposer une loi normale, même si rien théoriquement ne justifie cette méthode). La distribution des résidus peut être obtenue par le code suivant :

```
> CL <- glm(Y~lig+col, family=quasipoisson)
> residus=residuals(CL,type="pearson")
> par(mfrow = c(1, 2))
> hist(residus,breaks=seq(-3,5,by=.5),col="light green",proba=TRUE)
> u=seq(-4,5,by=.01)
> densite=density(residus)
> lines(densite,col="blue",lwd=1.5)
> lines(u,dnorm(u,mean(residus),sd(residus)),lty=2,col="red")
> plot(ecdf(residus))
> lines(u,pnorm(u,mean(residus),sd(residus)),lty=2,col="red")
> Femp=cumsum(densite$y)/sum(densite$y)
> lines(densite$x,Femp,,col="blue",lwd=1.5)
```

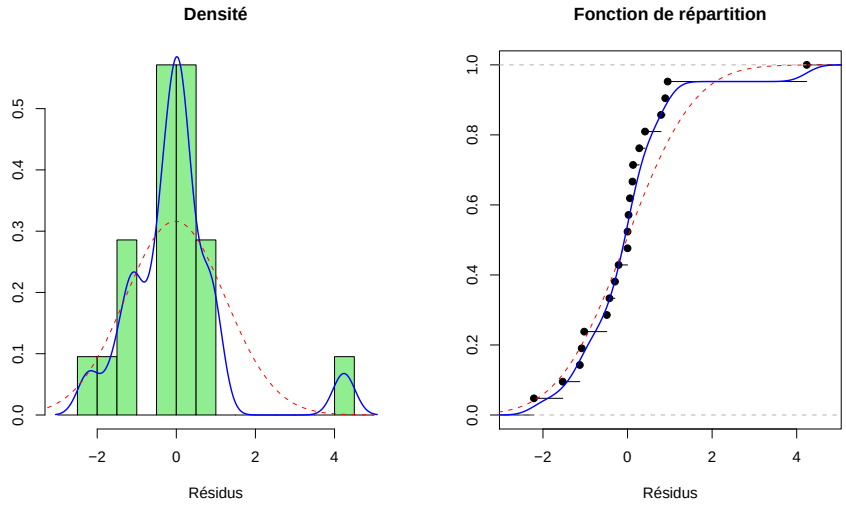


FIGURE 2.7 – Histogramme et densité des résidus (à gauche) et fonctions de répartition (à droite), avec l'ajustement Gaussien en pointillés.

Les triangles obtenus peuvent être visualisés sur la figure 2.4.4

Une fois simulé un pseudo triangle d'incréments de paiments, on prédit un montant de provision $\widehat{R}^b$ (par exemple via une méthode Chain Ladder). La variance des $\widehat{R}^b$ correspond à l'erreur d'estimation.

	0	1	2	3	4	5
0	0.948	-1.128	-1.533	-0.489	-0.427	0.000
1	0.024	0.277	-2.213	0.792	0.414	
2	0.116	0.056	-1.024	-0.297		
3	-1.082	0.891	4.237			
4	0.130	-0.211				
5	0.000					

TABLE 2.8 – Le triangle des résidus de Pearson, $\widehat{\varepsilon}_{i,j} = \widehat{\mu}_{i,j}^{-1/2} \cdot [Y_{i,j} - \widehat{\mu}_{i,j}]$.

Afin de prendre en compte l'erreur de modèle, plusieurs méthodes peuvent être utilisées. La première, et la plus simple, consiste à noter qu'à partir du pseudo triangle $Y_{i,j}^b$, peut obtenir des prédictions pour la partie inférieure, $\widehat{Y}_{i,j}^b$. Compte tenu du modèle Poissonnien, on peut alors simuler une trajectoire possible d'incréments de paiements en simulant les $Y_{i,j}^b$ à l'aide de loi de Poisson de paramètre $\widehat{Y}_{i,j}^b$. Le code est alors le suivant

```
> CLsimul1=function(triangle){
+  triangles=rpoisson(length(triangle),lambda=triangle)
+  return(sum(ULT-DIAG)) }
```

La seconde méthode est d'utiliser une relecture du modèle de [20], proposée par [8]. A partir du pseudo triangle, on va utiliser les facteurs de développement $\widehat{\lambda}_j$ et les variances associés $\widehat{\sigma}_j^2$ obtenus sur le triangle initial. On prolonge alors le triangle dans la partie inférience via le modèle dynamique

$$\widehat{C}_{i,j+1}^b | \widehat{C}_{i,j}^b \sim \mathcal{N}(\widehat{\lambda}_j \widehat{C}_{i,j}^b, \widehat{\sigma}_j^2 \widehat{C}_{i,j}^b).$$

Le code est alors le suivant, où `triangle` est un triangle de paiements *cumulés*, `l` correspond à un vecteur de facteurs de développement, et `s` à un vecteur de volatilités,

```
> CLsimul2=function(triangle,l,s){
+ m=nrow(triangle)
+ for(i in 2:m){
+  triangle[(m-i+2):m,i]=rnorm(i-1,
+            mean=triangle[(m-i+2):m,i-1]*l[i-1],
+            sd=sqrt(triangle[(m-i+2):m,i-1])*s[i-1])
+ }
+ ULT=triangle[,m]
+ DIAG=diag(triangle[,m:1])
+ return(sum(ULT-DIAG)) }
```

2.4.5 Le modèle binomial-négative

Ce modèle a été proposé par [31]. On suppose ici que

$$\mathbb{E}(X_{i,j}|\mathcal{F}_{j-1}) = [\lambda_{j-1} - 1] \cdot C_{i,j-1}$$

$$\text{Var}(X_{i,j}|\mathcal{F}_{j-1}) = \lambda_{j-1}[\lambda_{j-1} - 1] \cdot C_{i,j-1}$$

Pour rappels, la régression binomiale négative se fait à l'aide de la fonction `glm.nb` de `library(MASS)`, ou l'option `model = "negbin"` dans la fonction `Zelig`.

2.4.6 Quel modèle de régression ?

Comme nous l'avons mentionné dans le premier chapitre, deux paramètres fondamentaux interviennent dans une régression linéaire généralisée,

- la *fonction lien*, qui lie la prédiction aux facteurs, ici $\widehat{Y}_{i,j} = \mathbb{E}(Y_{i,j}|\mathcal{F}_n) = \exp[\widehat{\gamma} + \widehat{\alpha}_i + \widehat{\beta}_j]$,
- la *loi* ou la *fonction variance*, qui donne la forme de l'intervalle de confiance, ici $\text{Var}(Y_{i,j}|\mathcal{F}_n) = \phi \cdot \mathbb{E}(Y_{i,j}|\mathcal{F}_n)$,

L'unique motivation du modèle précédent est qu'il permet d'obtenir exactement le même montant que la méthode Chain Ladder. Mais aucun critère statistique n'a été évoqué, pour l'instant, afin de légitimer ce modèle.

Les modèles Tweedie sont une famille de *sur*modèle, incluant le modèle Poissonnien. On suppose que

- la *fonction lien*, est une fonction puissance, ou plutôt une tranformée de Box-Cox, $\widehat{Y}_{i,j} = g_\lambda^{-1}[\widehat{\gamma} + \widehat{\alpha}_i + \widehat{\beta}_j]$ où $g_\lambda(x) = \lambda^{-1}[x^\lambda - 1]$ si $\lambda > 0$ avec le cas limite $g_0(x) = \log(x)$.
- la *fonction variance*, qui donne la forme de l'intervalle de confiance, ici $\text{Var}(Y_{i,j}|\mathcal{F}_n) = \phi \cdot \mathbb{E}(Y_{i,j}|\mathcal{F}_n)^\mu$

où les paramètres λ et μ sont inconnus.

La densité[1] d'une loi Tweedie de paramètre `mu` est ici

```
> ftweedie = function(y,p,mu,phi){
+ if(p==2){f = dgamma(y, 1/phi, 1/(phi*mu))} else
+ if(p==1){f = dpois(y/phi, mu/phi)} else
+ {lambda = mu^(2-p)/phi /(2-p)
+ if(y==0){ f = exp(-lambda)} else
+ { alpha = (2-p)/(p-1)
+    beta = 1 / (phi * (p-1) * mu^(p-1))
+    k = max(10, ceiling(lambda + 7*sqrt(lambda)))
+    f = sum(dpois(1:k,lambda) * dgamma(y,alpha*(1:k),beta))
+ }}
+ return(f)}
```

1. où le terme *densité* s'entend au sens large, à savoir une probabilité dans le cas discret.

Afin de juger de la pertinance de l'ajustement, on peut calculer la log-vraisemblance du modèle, en gardant un lien logarithmique par exemple (ce qui est parfois plus simple au niveau numérique, mais aussi au niveau de l'interprétation),

```
> pltweedie <- function(puissance){
+ regt = glm(Y~lig+col, tweedie(puissance,0))
+ reserve = sum(fitted.values(regt)[!passe])
+ dev = deviance(regt)
+ phi.hat = dev/n
+ mu = fitted.values(regt)[passe]
+ hat.logL = 0
+ for (k in 1:length(Y)){
+    hat.logL <- hat.logL + log(ftweedie(Y[k], puissance, mu[k], phi.hat)) }
+  cat("Puissance =", round(puissance,3), "phi =", round(phi.hat,2),
+  "Reserve (tot) =", round(reserve), "logL =", round(hat.logL,3),"\n") }
```

Si on calcule la log-vraisemblance pour 5 valeurs, comprises entre 1 et 2 (correspondant respectivement au cas d'une régression Poisson et une régression Gamma), on obtient

```
> library(statmod)
> for(puissance in c(1,1.25,1.5,1.75,2)){pltweedie(puissance)}
Puissance = 1 phi = 166.95 Reserve (tot) = 1345 logL = -Inf
Puissance = 1.25 phi = 42.92 Reserve (tot) = 1216 logL = -151.72
Puissance = 1.5 phi = 15.8 Reserve (tot) = 996 logL = -145.232
Puissance = 1.75 phi = 9.02 Reserve (tot) = 609 logL = -153.997
Puissance = 2 phi = 6.78 Reserve (tot) = 125 logL = -170.614
```

La Figure 2.8 permet de visualiser l'influence du paramètre de la puissance de la fonction variance. La Figure montre aussi l'évolution du montant de provision $\widehat{R}$,

Si l'on souhaite garder un lien logarithmique, le paramètre le plus vraisemblance pour la fonction variance est entre 1 et 2, $\widehat{\mu} = 1.38$,

```
>  optimize(pltweedie, c(1.01,1.99),  tol=1e-4,maximum = TRUE)
Puissance = 1.384 phi = 23.78 Reserve (tot) = 1114 logL = -144.902
```

2.5 Les triangles multivariés

Comme nous l'avions expliqué dans l'introduction, l'utilisation des triangles, et des méthodes de cadences de paiements, n'est possible que si les triangles sont stables, et homogènes. Or il n'est pas rare qu'un triangle comporte des risques relativement différents dans leur développement. Par exemple en assurance auto, les accidents matériels et corporels sont sensiblement différents.

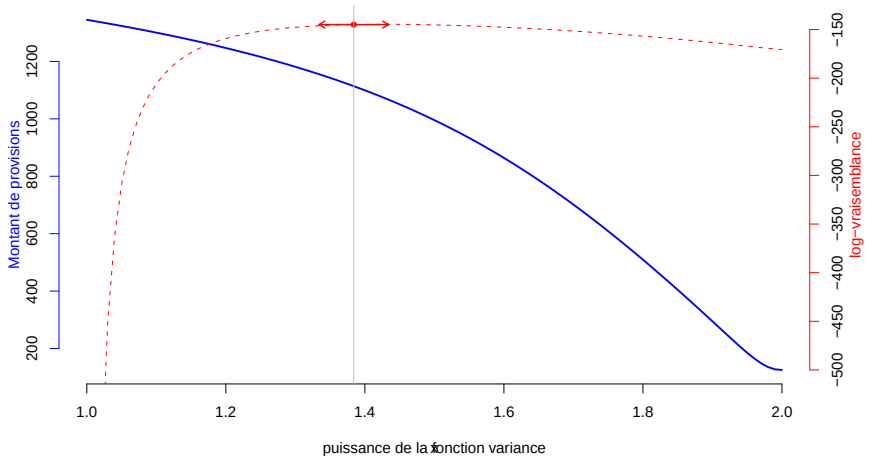


FIGURE 2.8 – Évolution de la log-vraisemblance profilée en fonction de μ et montant de provision $\widehat{R}$ estimé par GLM (avec un lien logarithmique).

2.5.1 Hypohtèse d'indépendance entre les triangles, et lois paramétriques

En s'insiprant de l'idée de [20], on peut supposer que $\widehat{R}_i$ suive une loi $LN(\mu_i, \sigma_i^2)$ pour $i = 1, 2$. Si l'on suppose les risques indépendant, la loi de la somme est simplement la convolée des deux lois. On peut utiliser les familles de distribution au format `S4` et la `library(distr)`. Rappelons que pour si $X \sim LN(\mu, \sigma^2)$,

$$\mu = \log[\mathbb{E}(X)] - \frac{1}{2}\log\left(1 + \frac{\text{Var}(X)}{\mathbb{E}(X)^2}\right) \text{ et } \sigma^2 = \log\left(1 + \frac{\text{Var}(X)}{\mathbb{E}(X)^2}\right).$$

A partir des moyennes et variances - données par la méthode de [20] par exemple - on en déduit les lois des deux montants de provision. Si on suppose que les deux triangles sont *indépendants*, alors

```
> library(distr)
> V=MackChainLadder(P.mat)$Total.Mack.S.E^2
> E=sum(MackChainLadder(P.mat)$FullTriangle[,n]-
+-diag(MackChainLadder(P.mat)$FullTriangle[n:1,]))
> mu = log(E) - .5*log(1+V^2/E^2)
> sigma2 = log(1+V^2/E^2)
> LM = Lnorm(meanlog=mu,sdlog=sqrt(sigma2))
> V=MackChainLadder(P.corp)$Total.Mack.S.E^2
> E=sum(MackChainLadder(P.corp)$FullTriangle[,n]-
```

```
+ diag(MackChainLadder(P.corp)$FullTriangle[n:1,]))
> mu = log(E) - .5*log(1+V^2/E^2)
> sigma2 = log(1+V^2/E^2)
> LC = Lnorm(meanlog=mu,sdlog=sqrt(sigma2))
> LT=LM+LC
```

On peut alors comparer la loi convolée, et la loi lognormale ajustée sur le triangle cumulé,

```
> P.tot = P.mat + P.corp
> library(ChainLadder)
> V=MackChainLadder(P.tot)$Total.Mack.S.E
> E=sum(MackChainLadder(P.tot)$FullTriangle[,n]-
+ diag(MackChainLadder(P.tot)$FullTriangle[n:1,]))
> mu = log(E) - .5*log(1+V^2/E^2)
> sigma2 = log(1+V^2/E^2)
> u=seq(0,qlnorm(.95,mu,sqrt(sigma2)),length=1000)
> vtotal=dlnorm(u,mu,sqrt(sigma2))
> vconvol=d(LT)(u)
> plot(u,vtotal)
> lines(u, vconvol)
```

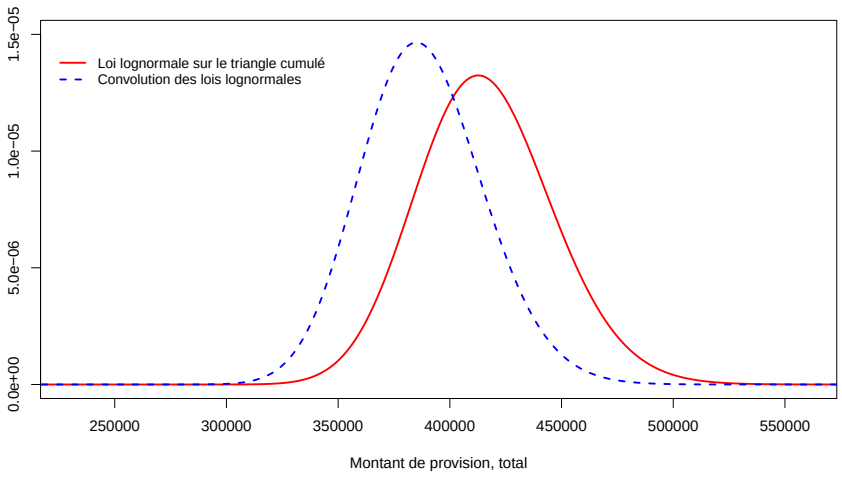


FIGURE 2.9 – Distribution du montant de provision $\widehat{R} = \widehat{R}_1 + \widehat{R}_2$.

Les quantiles à 95% sont alors respectivement

```
> cat("Quantile convolée =",q(LT)(.95))
Quantile convolée = 434615.9
```

```
> cat("Quantile lognormal =",qlnorm(.95,mu,sqrt(sigma2)))
Quantile lognormal = 467686.8
```

pour la loi convolée et pour la somme des deux triangles. Deux interprétations sont alors possibles : supposer les triangles comme étant indépendants est probablement une hypothèse trop forte et travailler sur un triangle agrégé (et donc peu homogène) introduit une incertitude supplémentaire.

2.5.2 Le modèle de Mack bivarié

[26] a proposé une méthode de type Chain-Ladder dans un cadre multivarié. On note

$$\boldsymbol{\lambda}_{i,j} = (\lambda_{i,j}^{(k)}) \text{ où } \lambda_{i,j}^{(k)} = \frac{C_{i,j}^{(k)}}{C_{i,j-1}^{(k)}}$$

et $\boldsymbol{C}_{i,j} = (C_{i,j}^{(k)}) \in \mathbb{R}^K$ On suppose qu'il existe $\boldsymbol{\lambda}_j = \in \mathbb{R}^K$

$$\mathbb{E}[\boldsymbol{C}_{i,j}|\boldsymbol{C}_{i,j-1}] = (\boldsymbol{\lambda}_{j-1}) \cdot \boldsymbol{C}_{i,j-1}$$

et

$$\text{Cov}[\boldsymbol{C}_{i,j}, \boldsymbol{C}_{i,j}|\boldsymbol{C}_{i,j-1}] = (\sqrt{\boldsymbol{C}_{j-1}}) \cdot \boldsymbol{\Sigma}_{j-1} \cdot (\sqrt{\boldsymbol{C}_{j-1}})$$

Alors sous ces hypothèses, comme dans le cas univarié, on peut écrire

$$\mathbb{E}[\boldsymbol{C}_{i,n}|\boldsymbol{C}_{i,n-i}] = \prod_{j=n-i}^{n-1} (\boldsymbol{\lambda}_j)\boldsymbol{C}_{i,n-i}.$$

L'estimateur du facteur de transition est

$$\widehat{\boldsymbol{\lambda}}_j = \left[\sum_{i=0}^{n-j-1} (\sqrt{i,j}) \cdot \boldsymbol{\Sigma}_j^{-1} \cdot (\sqrt{i,j}) \right]^{-1} \cdot \sum_{i=0}^{n-j-1} (\sqrt{i,j}) \cdot \boldsymbol{\Sigma}_j^{-1} \cdot (\sqrt{i,j}) \boldsymbol{\lambda}_{i,j+1}$$

L'estimateur Chain-Ladder de la charge ultime est

$$\widehat{\boldsymbol{C}_{i,n}} = \prod_{j=n-i}^{n-1} (\widehat{\boldsymbol{\lambda}}_j)\boldsymbol{C}_{i,n-i}.$$

Cet estimateur vérifie les mêmes propriétés que dans le cas univarié. En particulier, cet estimateur est un estimateur sans biais de $\mathbb{E}[\boldsymbol{C}_{i,n}|\boldsymbol{C}_{i,n-i}]$ mais aussi de $\mathbb{E}[\boldsymbol{C}_{i,n}]$.

Il est aussi possible de calculer les mse de prédiction.

2.5.3 Modèles économétriques pour des risques multiples

L'idée dans les modèles économétriques est de supposer que les *résidus* peuvent être corrélés,

```
>  ligne = rep(1:n, each=n); colonne = rep(1:n, n)
>  passe = (ligne + colonne - 1)<= n
>  PAID=P.corp; INC=PAID
>  INC[,2:n]=PAID[,2:n]-PAID[,1:(n-1)]
>  I.corp = INC
>  PAID=P.mat; INC=PAID
>  INC[,2:n]=PAID[,2:n]-PAID[,1:(n-1)]
>  I.mat = INC
>  Ym = as.vector(I.mat)
>  Yc = as.vector(I.corp)
>  lig = as.factor(ligne)
>  col = as.factor(colonne)
>  base = data.frame(Ym,Yc,col,lig)
>  regm=glm(Ym~col+lig,data=base,family="poisson")
>  regc=glm(Yc~col+lig,data=base,family="poisson")
>  res.corp=residuals(regc,type="pearson")
>  res.mat=residuals(regm,type="pearson")
>  plot(res.corp,res.mat)
```

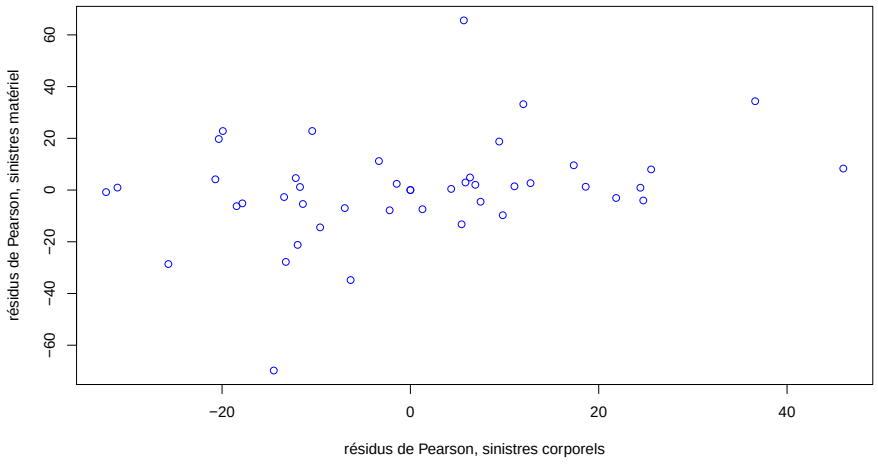


FIGURE 2.10 – Nuage de points de résidus de Pearson, obtenus sur un modèle log-Poisson, $(\widehat{\varepsilon}_{i,j}^{\text{matériel}}, \widehat{\varepsilon}_{i,j}^{\text{corporel}})$.

On notera que la corrélation n'est pas nulle.

```
> cat("Corrélation des résidus =",cor(res.mat,res.corp))
Corrélation des résidus = 0.2957895
```

Une fois notée qu'il existe probablement une dépendance entre les deux triangles, il semble légitime de la prendre en compte dans les algorithmes de simulations évoqués dans la partie 2.4.4.

- pour l'erreur d'estimation, quand on tire les résidus, on ne les tire pas indépendement dans les deux triangles. On tire alors les *paires* de résidus $(\widehat{\varepsilon}_{i,j}^{\text{matériel},b}, \widehat{\varepsilon}_{i,j}^{\text{corporel}}, b)$
- pour l'erreur, on peut tirer une loi de Poisson bivariée si on utilise une régression Poissonnienne bivariée (implémentée dans `library()bivpois` ou un vecteur Gaussien bivarié.

Dans le second cas,

$$\begin{pmatrix} C_{i,j+1}^{\text{matériel}} \\ C_{i,j+1}^{\text{corporel}} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \lambda_j^m C_{i,j}^{\text{matériel}} \\ \lambda_j^c C_{i,j}^{\text{corporel}} \end{pmatrix}, \begin{pmatrix} \sigma_j^{m2} C_{i,j}^{\text{matériel}} & \star \\ \star & \sigma_j^{c2} C_{i,j}^{\text{corporel}} \end{pmatrix} \right)$$

2.6 Borhutter-Fergusson, Benktander et les méthodes bayésiennes

Les deux premières méthodes que nous allons voir ont souvent été proposées comme une alternative à la méthode Chain Ladder, car elles introduisent un *a priori* sur la charge ultime.

2.6.1 Le modèle de Borhutter-Ferguson et l'introduction d'un avis d'expert

Classiquement, on continue ici à supposer que

- les années de survenance sont indépendantes les unes des autres
- il existe μ_i et des facteurs de développement $\beta_1, \beta_2, \cdots, \beta_n$ - avec $\beta_n = 1$ - tels que

$$\mathbb{E}(C_{i,1}) = \beta_1 \mu_i$$

$$\mathbb{E}(C_{i,j+k}|C_{i,1}, \cdots, C_{i,j}) = C_{i,j} + [\beta_{j+k} - \beta_j]\mu_i$$

Sous ces hypothèses, pour tout i, j $\mathbb{E}(C_{i,j}) = \beta_j \mu_i$. Ce qui peut rappeler les modèles à facteurs évoqués auparavant. Sauf qu'ici, seul $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_n)$ sera à estimer statistiquement, $\boldsymbol{\mu} = \widehat{\mu}_i$ étant obtenu par *avis d'expert*, $\widehat{\mu}_i$ étant un estimateur de $\mathbb{E}(C_{i,n})$. Moyennant ces deux estimations, on en déduit l'estimateur de $\mathbb{E}(C_{i,n}|C_{i,1}, \cdots, C_{i,j})$ de la forme

$$\widehat{C}_{i,n} = C_{i,j} + [1 - \widehat{\beta}_{j-i}]\widehat{\mu}_i$$

L'estimateur proposé par Bornhutter-Ferguson est alors simplement obtenu à partir de la méthode Chain-Ladder, en posant

$$\widehat{\beta}_j = \prod_{k=j+1}^{n} \frac{1}{\widehat{\lambda}_k}$$

Enfin, pour estimer $\widehat{\mu}_i$, on suppose disposer d'un ratio sinistre/prime cible, par exemple de 105%, par année de survenance. Dans ces conditions, on peut alors estimer simplement le montant de provision,

```
> mu <- 1.05*PREMIUM
> beta <- rev(cumprod(rev(1/LAMBDA)))
> Cdiag = diag(PAID[,n:1])
> Cultime <- Cdiag+(1-c(1,rev(beta)))*mu
> cat("Total reserve =",Cultime-Cdiag)
Total reserve =   0.00   23.12   33.49   58.98  131.26 1970.45
```

i	0	1	2	3	4	5
prime	4591	4692	4863	5175	5673	6431
$\widehat{\mu}_i$	4821	4927	5106	5434	5957	6753
λ_i	1,380	1,011	1,004	1,002	1,005	
β_i	0,708	0,978	0,989	0,993	0,995	
$\widehat{C}_{i,n}$	4456	4753	5453	6079	6925	7187
$\widehat{R}_i$	0	23	33	59	131	1970

TABLE 2.9 – Estimation du montant de provision par Borhutter-Ferguson, avec un ratio sinistres/primes de 105%.

2.6.2 Benktander

L'estimateur de [3], repris quelques années plus tard par [14], repose sur un estimateur *a priori* de la charge ultime $C_{i,n}$, noté μ_i. On suppose également qu'il existe une cadence de paiemments $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_n)$, connue, telle que

$$\mathbb{E}(C_{i,j}) = \mu_i \beta_j$$

Sous ces hypothèses, le montant de provision devrait être

$$\widehat{R}_i = \widehat{C}_{i,n} - C_{i,n-i} = [1 - \beta_{n-i}]\mu_i$$

Au lieu de se baser uniquement sur μ_i, [3] avait proposé un estimateur crédibilisé de la charge ultime, de la forme

$$\beta_{n-i}\widehat{C}_{i,n}^{\text{CL}} + [1 - \beta_{n-i}]\mu_i$$

Il s'agit d'utiliser l'estimateur Chain-Ladder, moyenné avec l'estimation a priori de la charge ultime. Alors

$$\widehat{R}_i^{\text{BH}} = \widehat{C}_{i,n} - C_{i,n-i} = [1-\beta_{n-i}]\left(\beta_{n-i}\widehat{C}_{i,n}^{\text{CL}} + [1-\beta_{n-i}]\mu_i\right)$$

On notera que

$$\widehat{R}_i^{\text{BH}} = (1-\beta_{n-i})\widehat{C}_i^{\text{BF}}$$

si la cadence $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_n)$ est construite à partir des facteurs de développement induits par la méthode Chain-Ladder. Une autre écriture de cette expression est d'écrire la charge ultime (et non plus le montant de provision),

$$\widehat{C}_i^{\text{BH}} = C_{i,n-i} + (1-\beta_{n-i})\widehat{C}_i^{\text{BF}} = \beta_{n-i}\widehat{C}_i^{\text{CL}} + (1-\beta_{n-i})\widehat{C}_i^{\text{BF}}$$

ce qui permet de voir la prédiction de Benktander comme une combinaison convexe des estimateurs Chain-Ladder et de Bornhuetter-Ferguson.

2.6.3 La méthode dite *Cape-Code*

Dans cette approche, on utilise là encore un avis d'expert. L'idée est de réécrire l'expression

$$C_{i,n} = C_{i,n-i} + \left(1 - \frac{C_{i,n-i}}{C_{i,n}}\right) C_{i,n}$$

sous la forme

$$C_{i,n} = C_{i,n-i} + \left(1 - \frac{C_{i,n-i}}{C_{i,n}}\right) LR_i \cdot P_i,$$

où LR_i correspond au *loss ratio* pour l'année i, i.e. $LR_i = C_{i,n}/P_i$. L'idée de la méthode dite *Cape-Code* est d'écrire une forme plus générale,

$$C_{i,n} = C_{i,n-i} + (1-\pi_{n-i})\, LR_i P_i$$

où π_{n-i} correspond à une cadence de paiement, et peut être estimé par la méthode Chain Ladder. Quant aux LR_i il s'agit des *loss ratio* cibles, correspondant à un avis d'expert. On peut aussi proposer un même ratio cible pour plusieurs années de survenance. On posera alors

$$R_i = C_{i,n} - C_{i,n-i} = (1-\pi_{n-i}) LR_{\mathcal{A}} P_i.$$

pour $i \in \mathcal{A}$, où

$$LR_{\mathcal{A}} = \frac{\sum_{k\in\mathcal{A}} C_{n,n-k}}{\sum_{k\in\mathcal{A}} \pi_{n-k} P_k}.$$

Dans un premier temps, on peut calculer les π_i à partir de la méthode Chain Ladder, i.e.

$$\pi_{n-i} = \frac{C_{i,n-i}}{C_{i,n}}$$

où la charge ultime est celle prédite pas la méthode Chain-Ladder.

```
> (PI <- 1-Cdiag/Cultime)
[1] 0.000000 0.004713 0.006559 0.010855 0.022036 0.291809
>  LR=TRIANGLE[,6]/PREMIUM
>  Cdiag <- diag(PAID[,n:1])
>  Cultime <- TRIANGLE[,6]
> cat("Coef. PI =", (Cultime-Cdiag)/(LR*PREMIUM))
Coef. PI = 0.00000 0.00471 0.00656 0.01085 0.02204 0.29181
```

Si on suppose ensuite que $\mathcal{A} = \{1, 2, \cdots, n\}$, alors

```
>  LR=sum(TRIANGLE[,6])/sum(PREMIUM)
> cat("Total reserve =",  PI*LR*PREMIUM)
Total reserve =   0.0000   24.5832   35.6120   62.7199  139.5729 2095.2682
```

On obtient ici un montant de provision total inférieur à celui obtenu par la méthode Chain Ladder puisque `sum(R)` vaut ici `2357.756`.

2.6.4 Les approches Bayésiennes

Les approches Bayésiennes ont été popularisées en sciences actuarielles par la théorie de la crédibilité, correspondant à une approche Bayésienne dans un cadre linéaire. Mais il est possible d'aller plus loin.

Classiquement, supposons que l'on s'intéresse à dont la loi serait $f(\cdot|\boldsymbol{\theta})$, où très généralement, $= (Y_{i,j})$ et $\boldsymbol{\theta} = (\theta_{i,j})$. peut être ici le triangle des paiements cumulés , le triangle des incréments , ou le triangle des coefficients de transition des cadences de paiements $\boldsymbol{\lambda} =_{i,j+1} /_{i,j}$.

Exemple 2.1 *Dans l'approche de [20],* $\boldsymbol{\theta}_j = (\lambda_j, \sigma_j^2)$.

Application aux cadences de paiements

Ici, on s'intéresse à la loi de $\boldsymbol{\lambda}$, qui dépendra de $\boldsymbol{\theta} = (\boldsymbol{\theta}_j)$ où $\boldsymbol{\theta}_j = (\gamma_j, \sigma_j^2)$, où, pour des simplicités de notations (et éviter de confondre avec les $\lambda_{i,j}$) on note γ_j le facteur de développement sous-jacent.

$$\lambda_{i,j}|(\gamma_j, \sigma_j^2) \sim \left(\gamma_j, \frac{\sigma_j^2}{C_{i,j}}\right)$$

Ici, σ^2 ne sont pas les paramètres d'intérêt, et sont supposés estimés séparément (comme nous le faisions déjà dans les modèles linéaires généralisés). Quant aux $C_{i,j}$, ils sont interprétés ici comme des poids, et sont supposés connus. La log-vraisemblance est ici

$$\log \mathcal{L}(\boldsymbol{\lambda}|\boldsymbol{\gamma}) = \sum_{i,j} \frac{1}{2} \left(\log \left[\frac{C_{i,j-1}}{\sigma_j^2} \right] - \frac{C_{i,j-1}}{\sigma_j^2} [\lambda_{i,j} - \gamma_j]^2 \right).$$

En utilisant la formule de Bayes, la log-densité de $\boldsymbol{\gamma}$ conditionnelle aux $\boldsymbol{\lambda}$ est simplement

$$\log[g(\boldsymbol{\gamma}|\boldsymbol{\lambda})] = \log[\pi(\boldsymbol{\gamma})] + \log[\mathcal{L}(\boldsymbol{\lambda}|\boldsymbol{\gamma})] + \text{constante},$$

où $\pi(\cdot)$ est une loi *a priori* de $\boldsymbol{\gamma}$ (par exemple un vecteur Gaussien).

L'algorithme de Gibbs et généralisations

On cherche ici à générer un ensemble de vecteurs aléatoires $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_m) \in \mathbb{R}^m$. Contrairement aux méthodes de Monte Carlo où l'on cherche à générer des vecteurs indépendants les uns des autres, on va essayer de construire une suite de manière récurente, vérifiant des propriétés d'ergodicité.

On part d'un vecteur initial $\boldsymbol{\gamma}^{(0)} = (\gamma_1^{(0)}, \cdots, \gamma_m^{(0)})$, par exemple les valeurs obtenues par la méthode Chain Ladder puis on génère, de manière itérée

$$\left\{\begin{array}{l}
\gamma_1^{(k+1)} \sim f(\cdot|\gamma_2^{(k)}, \cdots, \gamma_m^{(k)}, \lambda) \\
\gamma_2^{(k+1)} \sim f(\cdot|\gamma_1^{(k+1)}, \gamma_3^{(k)}, \cdots, \gamma_m^{(k)}, \lambda) \\
\gamma_3^{(k+1)} \sim f(\cdot|\gamma_1^{(k+1)}, \gamma_2^{(k+1)}, \gamma_4^{(k)}, \cdots, \gamma_m^{(k)}, \lambda) \\
\vdots \\
\gamma_{m-1}^{(k+1)} \sim f(\cdot|\gamma_1^{(k+1)}, \gamma_2^{(k+1)}, \gamma_{m-2}^{(k+1)}, \gamma_m^{(k)}, \lambda) \\
\gamma_m^{(k+1)} \sim f(\cdot|\gamma_1^{(k+1)}, \gamma_2^{(k+1)}, \cdots, \gamma_{m-1}^{(k+1)}, \lambda)
\end{array}\right.$$

Ces lois conditionnelles n'ayant pas forcément de forme simple, l'algorithme de *metropolis* (d'acceptation-rejet) peut alors être utiliser pour simuler ces différentes lois conditionnelle.

La *méthode de rejet* est basé sur l'idée suivante

- on souhaite tirer (indépendemment) suivant une loi f, qu'on ne sait pas simuler
- on sait simuler suivant une loi g qui vérifie $f(x) \leq Mg(x)$, pour tout x, où M peut être calculée.

L'agorithme pour tirer suivant f est alors le suivant

- répéter
  - tirer Y selon la loi g
  - tirer U selon la loi uniforme sur $[0,1]$, indépendamment de Y,
- tant que $U > \dfrac{f(Y)}{Mg(Y)}$.
- poser $X = Y$.

Exemple 2.1 *On peut utiliser cette technique pour simuler une loi normale à partir d'une loi de Laplace (qui est une variable exponentielle avec un signe positif ou négatif), de densité* $g(x) = 0.5 \cdot \exp(-|x|)$, *avec* $M = \sqrt{2e\pi^{-1}}$. *Mais cet algorithme est très coûteux en temps s'il y a beaucoup de rejets,*

comme le montre le code suivant : ici, on note que l'on perd près du quart des simulations.

```
> n=1000
> Y=rexp(n)* sample(c(-1,1),size=n,replace=TRUE)
> U=runif(n)
> dlaplace=function(x){.5*exp(-abs(x))}
> M=sqrt(2*exp(1)/pi)
> test=U<dnorm(Y)/(M*dlaplace(Y))
> mean(test)
[1] 0.761
> X=Y[test==TRUE]
> plot(density(X))
> curve(dnorm(x),add=TRUE,col="red",lty=2,xlab="")
```

FIGURE 2.11 – Simulation d'une variable Gaussienne par une méthode de rejet basée sur une loi de Laplace.

L'*adaptative rejection sampling* est une extension de cet algorithme, à condition d'avoir une densité *log-concave*. On parle aussi de *méthode des cordes*. On majore localement la fonction $\log f$ par des fonctions linéaires, autrement dit, on construit alors une enveloppe à $\log f$. On majore alors f par une fonction g_n constituées de n fonctions linéaires par morceaux, comme le montre la figure 2.12

Formellement, on construit $L_{i,j}(x)$ la droite reliant les points $(x_i, \log(f(x_i)))$ et $(x_j, \log(f(x_j)))$. On pose alors

$$h_n(x) = \min\left\{L_{i-1,i}(x), L_{i+1,i+2}(x)\right\},$$

qui définie alors une enveloppe de $\log(f)$ (par concavité de $\log(f)$. On utilise alors un algorithme de rejet avec comme fonction de référence

$$g_n(x) = \frac{\exp(h_n(x))}{\int \exp(h_n(t))dt} \text{ normalisée pour définir une densité.}$$

L'algorithme est alors le même que précédemment, à savoir

- répérer
  - tirer Y selon la loi g_n
  - tirer U selon la loi uniforme sur $[0,1]$, indépendamment de Y,
- tant que $U > \dfrac{f(Y)}{\exp(h_n(Y))}$.
- poser $X = Y$.

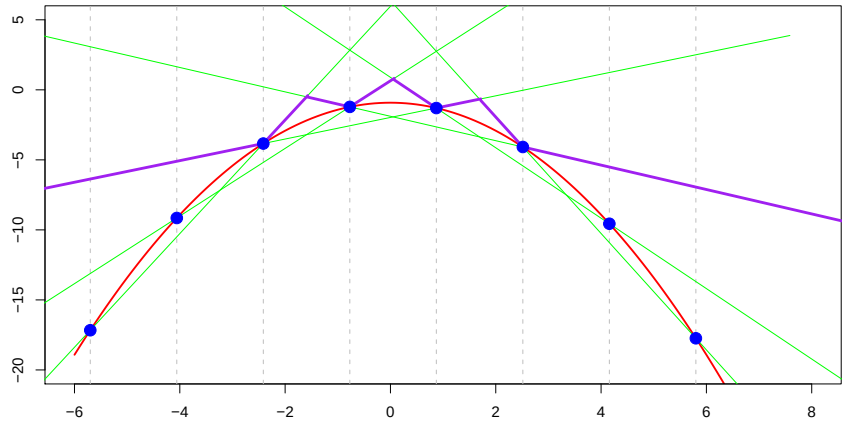


FIGURE 2.12 – Majoration d'une log-densité par des fonctions linéaires par morceaux.

Enfin, dans le cas des densités non log-concave, il est possible de rajouter une étape. En effet, dans la construction précédente, la fonction h_n est toujours un majorant, mais ce n'est plus forcément une enveloppe de $\log(f)$. Il suffit alors de rajouter une étape de rejet supplémentaire.

Le plus simple est d'implémenter un algorithme de Gibbs, c'est à dire créer une suite de variables $X_1, X_2, \cdots$ par un processus itératif, Markovien. Les variables ne sont plus indépendantes, mais en invoquant des résultats d'ergodicité, les calculs de moyennes, de quantiles, ou de lois marginales restent valides.

Supposons que l'on dispose de X_{k-1}. Pour tirer X_k, on utilise l'algorithme précédant, et la nouvelle étape de rejet est la suivante

- tirer U selon la loi uniforme sur $[0,1]$, indépendamment de X et de X_{k-1},
  - si $U > \min\left\{1, \dfrac{f(X)\min\{f(X_{k-1}), \exp(h_n(X_{k-1}))\}}{f(X_{k-1})\min\{f(X), \exp(h_n(X))\}}\right\}$ alors garder $X_k = X_{k-1}$
  - sinon poser $X_k = X$

Ces fonctions exponentielles par morceaux sont inéressantes car elles sont faciles à simuler. La fonction h_n est linéaires par morceaux, avec comme noeuds N_k, de telle sorte que

$$h_n(x) = a_k x + b_k \text{ pour tout } x \in [N_k, N_{k+1}].$$

Alors $g_n(x) = \dfrac{\exp(h_n(x))}{I_n}$ où

$$I_n = \int \exp(h_n(t))dt = \sum \frac{\exp[h_n(N_{k+1})] - \exp[h_n(N_k)]}{a_k}.$$

On calcule alors G_n, la fonction de répartition associée à g_n, et on fait utilise une méthode d'inversion pour tirer suivant G_n.

2.6.5 Approche bayésienne sur les facteurs de développement

En s'inspirant de la relecture du modèle de [20],

$$\widehat{C}^b_{i,j+1} | \widehat{C}^b_{i,j} \sim \mathcal{N}(\widehat{\lambda}_j \widehat{C}^b_{i,j}, \widehat{\sigma}^2_j \widehat{C}^b_{i,j}).$$

nous pouvons supposer que les facteurs de développements $\lambda_{i,j}$ suivent une loi lognormale, comme le suggérait [2].

La Figure 2.13 montre la simulation de 1000 valeurs $\widehat{R}$ de montant de provision, par cette méthode. La série n'est pas i.i.d. mais elle vérifie des propriétés d'ergodicité qui autorisent en particulier l'étude de la distribution du montant de provision.

Pour implémenter la méthode Bayésienne, il faut commencer par définir la fonction de log-vraisemblance associé au vecteur de facteurs de transition, noté ici `g`,

```
> Log_Ln<-function(g,C=PAID,sigma2=SIGMA^2,f=PAID[,2:n]/PAID[,1:(n-1)])
+ {s=0
+ for(i in 1:(n-1)){
+ for(j in 1:(n-i)){
+ s<-s+0.5*(log(C[i,j]/sigma2[j])-(C[i,j]/sigma2[j])*
+ (f[i,j]-exp(g[j]))^2)}}
+ return(s)}
```

Commençons par initialiser les valeurs,

```
gamma<-log(LAMBDA)
Mat_parametres<-gamma
parametres<-gamma
n_it<-1
```

On fait alors une boucle définie par un `while(n_it <= nb_iteration_gibbs*length(parametres))`. A chaque itération de la boucle, on commence par initialiser la valeur à échantillonner,

```
rg_var<-n_it-trunc(n_it/length(parametres))*length(parametres)+1
Xcur<-parametres[rg_var]
n_seg<-20
```

On crée alors la segmentation de l'intervalle,

```
t1<-colSums(f+C[,1:ncol(C)-1]*matrix(rep(sigma2,nrow(f)),nrow(f),ncol(f),
byrow = TRUE)-f,na.rm =TRUE)
t2<-(colSums(f+C[,1:ncol(C)-1]-f,na.rm =TRUE))^2
V_gamma<-1/lambda^2*t1/t2
X_inf<-gamma-coef*sqrt(V_gamma)
X_sup<-gamma+coef*sqrt(V_gamma)
X<-seq(0,n_seg-1)
X<-X_inf[rg_var]+X/(n_seg-1)*(X_sup[rg_var]-X_inf[rg_var])
X<-sort(X)
```

La première étape importante de l'algorithme est de simuler un vecteur `Xu` à partir de `gn`,

```
step1<-FALSE
while(step1==FALSE){
lg_vrais<-function(x,rg_var.){
param<-parametres
param[rg_var.]<-x
Log_Ln(param)
}
y<-lg_vrais(X[1],rg_var)
for(k in 2:n_seg)y<-cbind(y,lg_vrais(X[k],rg_var))
Y<-y[1,]
a=(Y[2:n_seg]-Y[1:(n_seg-1)])/(X[2:n_seg]-X[1:(n_seg-1)])
b=Y[1:(n_seg-1)]-a*X[1:(n_seg-1)]
X_int=-(b[3:length(b)]-b[1:(length(b)-2)])/(a[3:length(b)]-b[1:(length(b)-2)])
Y_int=a[1:(length(b)-2)]*X_int+b[1:(length(b)-2)]
X2=X
Y2=Y
for (k in 1:length(X_int)){
if(X_int[k]>=X[k+1] & X_int[k]<=X[k+2])
{ if(Y_int[k]>=a[k+1]*X_int[k]+b[k+1]){
X2<-c(X2,X_int[k])
Y2<-c(Y2,Y_int[k])
}}}
M<-rbind(X2,Y2)
M<-M[,order(M[1,])]
X2<-M[1,]
Y2<-M[2,]
a2=(Y2[3:(length(Y2)-1)]-Y2[2:(length(Y2)-2)])/(X2[3:(length(Y2)-1)]
-X2[2:(length(Y2)-2)])
b2=Y2[2:(length(Y2)-2)]-a2*X2[2:(length(Y2)-2)]
if(Y[1]>=a[2]*X[1]+b[2]){
```

```
a2<-c(a[1],a2)
b2<-c(b[1],b2)}
else {
a2<-c(a[2],a2)
b2<-c(b[2],b2)}
if(Y[n_seg]>=a[n_seg-2]*X[n_seg]+b[n_seg-2]){
a2<-c(a2,a[n_seg-1])
b2<-c(b2,b[n_seg-1])}
else {
a2<-c(a2,a[n_seg-2])
b2<-c(b2,b[n_seg-2])}
temp<-(exp(Y2[2:length(Y2)])-exp(Y2[1:length(Y2)-1]))/a2
m_n=sum(temp)
G_n=rep(0,length(X2))
for(k in 1:length(temp)){
G_n[k+1]<-G_n[k]+temp[k]/m_n}
U<-runif(1,0,1)
k_Xu<-1
while(G_n[k_Xu+1]<=U) k_Xu<-k_Xu+1
Xu<-1/a2[k_Xu]*(log((U-G_n[k_Xu])*m_n*a2[k_Xu]+exp(Y2[k_Xu]))-b2[k_Xu])
```

Pour faire un algorithme de rejet, on tire ensuite une loi uniforme `U<-runif(1,0,1)`, et on peut ensuite lancer la seconde étape de l'algorithme

```
val<-exp(lg_vrais(Xu,rg_var))/exp(a2[k_Xu]*Xu+b2[k_Xu])
lg_vrais(Xu,rg_var)
if (U>val){
n_seg<-n_seg+1
X<-c(X,Xu)
sort(X)}
else{
Xa=Xu
step1<-TRUE
}}
```

La troisième étape est l'algorithme d'Hastings-Metropolis, autrement dit on pose `U<-runif(1,0,1)`, puis la procédure est la suivante

```
k_Xcur<-1
while(X2[k_Xcur+1]<=Xcur& k_Xcur+1<=length(X2)) k_Xcur<-k_Xcur+1
k_Xa<-1
while(X2[k_Xa+1]<=Xa & k_Xa+1<=length(X2)) k_Xa<-k_Xa+1
num<-exp(lg_vrais(Xa,rg_var))*min(exp(lg_vrais(Xcur,rg_var))
,exp(a2[k_Xcur]*Xcur+b2[k_Xcur]))
denom<-exp(lg_vrais(Xcur,rg_var))*min(exp(lg_vrais(Xa,rg_var))
```

```
,exp(a2[k_Xa]*Xa+b2[k_Xa]))
if (U>min(1,num/denom)){
Xm<-Xcur }else{Xm<-Xa}
parametres[rg_var]<-Xm
if(rg_var==length(parametres)) Mat_parametres<-rbind(Mat_parametres,parametres)
n_it<-n_it+1
```

Ce qui constitue la fin de la boucle. On dispose alors d'une matrice de paramètres `Mat_parametres`, que l'on va utiliser pour obtenir un vecteur de montant de provisions par année de survenance,

```
for (k in 1:nrow(Mat_parametres)){
lambda_hat<-exp(Mat_parametres[k,])
Mat_prov<-C
for(l in (n+2):(n+n)){
i_dep<-l-n
for (i in i_dep:n){
a<-lambda_hat[l-i-1]^2*Mat_prov[i,l-i-1]/sigma2[l-i-1]
s<-sigma2[l-i-1]/lambda_hat[l-i-1]
Mat_prov[i,l-i]<-rgamma(1,shape = a,scale = s )}
}
last<-rep(0,n)
for (i in 1:n)
last[i]<-Mat_prov[i,n+1-i]
Res<-t(Mat_prov[,n])-last
if(k==1){Mat_res<-cbind(Res,sum(Res))}
if(k>1){Mat_res<-rbind(Mat_res,cbind(Res,sum(Res)))}
}
```

Toute l'information est alors dans le vecteur `Mat_res`. Un ensemble de simulations du montant total apparait dans la dernière ligne de `Mat_res`.

Tout le code précédant peut être obtenu à l'aide de la commande `source(bayes-triangle)`.

```
> source(bayes-triangle)
> plot(B$reserves[,6])
> abline(h=2426.88)
> apply(B$reserves,2,mean)
[1]    0.00000   22.46982   35.78285   66.30972  152.50063 2150.45450 2427.51752
```

La Figure 2.14 montre ainsi la distribution du montant de provision obtenu par cet algorithme, ainsi que les ordres de grandeurs des quantiles à 95%, 99% et 99.5%.

En iterant cette fonction, on peut d'ailleurs noter que l'estimation du quantile à 95% est relativement robuste, avec 10 000 tirages.

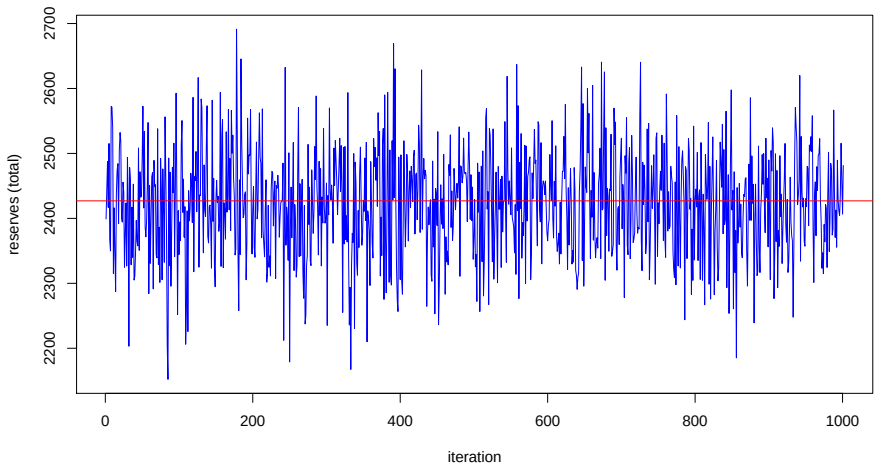


FIGURE 2.13 – Génération d'une suite de montants de provisions $\widehat{R}$.

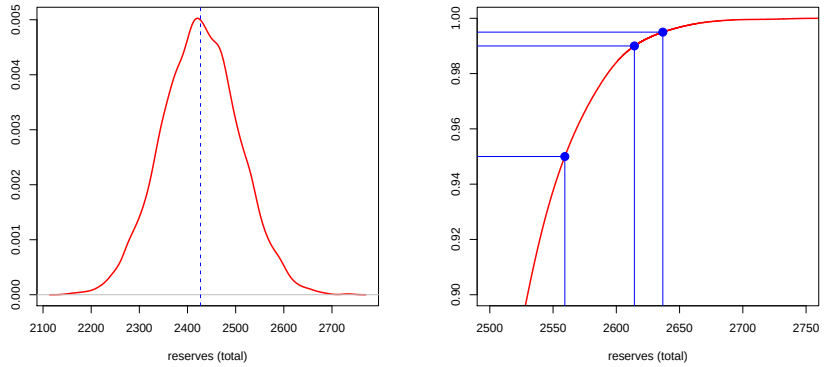


FIGURE 2.14 – Distribution du montants de provisions $\widehat{R}$, et estimation du quantile à 95%.

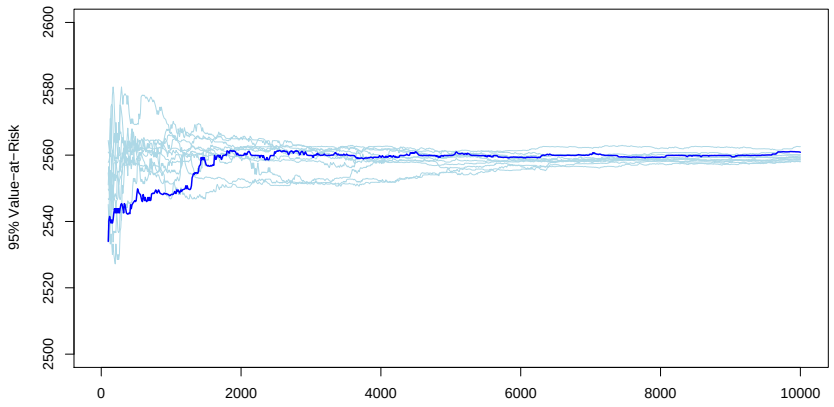


FIGURE 2.15 – Convergence du quantile à 95% du montant de provision.

Bibliographie

[1] R.A. Bailey. Insurance rates with minimum bias. *Proceedings of the Society of Actuaries*, 50 :4–11, 1963.

[2] N. Balson. *Mesure d'incertitude sur l'estimation des provisions de sinistres en Assurance Non Vie.* Institut des Actuaires - ENSAE, 2008.

[3] G. Benktander. An approach to credibility in calculating ibnr for casualty excess reinsurance. *Actuarial Review*, 3 :7–31, 1976.

[4] S. Christofides. Regression models based on log-incremental payments. In Institute of Actuaries, editor, *Claims Reserving Manual*, 1989.

[5] A.C. Davison and E.J. Snell. Residuals and diagnostics. In N. Reid D.V. Hinkley and E.J. Snell, editors, *Statistical Theory and Modelling.* Chapman and Hall.

[6] P. de Jong and G.H. Zeller. *Generalized Linear Models for Insurance Data.* Cambridge University Press, 2008.

[7] M. Denuit and A. Charpentier. *Mathématiques de l'assurance non-vie : Tarification et provisionnement. Tome 2.* Economica, 2005.

[8] P. D. England and R. J. Verrall. Analytic and bootstrap estimates of prediction errors in claims reserving. *Insurance : Mathematics and Economics*, 25 :281–293, 1999.

[9] E.W. Frees. *Regression modeling with actuarial and financial applications.* Cambridge University Press, 2009.

[10] J. Friedman. Multivariate additive regression splines. *Annals of Statistics*, 19(1) :1–67, 1991.

[11] H.U. Gerber and E.S.W. Shiu. Option pricing by esscher transforms. *Transactions of the Society of Actuaries Society of Actuaries*, 46 :99–191, 1994.

[12] C. A. Hachemeister and J. N. Stanard. Ibnr claims count estimation with static lag functions. In *12th ASTIN Colloquium*, Portimao, Portugal, 1975.

[13] T. Hastie and R. Tibshirani. *Generalized Additive Models.* Chapman and Hall, 1990.

[14] E. Hovinen. Additive and continuous ibnr. In *ASTIN Colloquium*, Loen, Norway, 1981.

[15] C. M. Hurvich and C.-L. Tsai. Model selection for extended quasi-likelihood models in small samples. *Biometrics*, 51 :1077–1084, 1995.

[16] J. Jung. On automobile insurance ratemaking. *ASTIN Bulletin*, 5 :41–48, 1968.

[17] M. ; Dhaene J. Kaas, R. ; Goovaearts and M. Denuit. *Modern Actuarial Risk Theory*. Springer Verlag, 2009.

[18] E. Kremer. Ibnr claims and the two-way model of anova. *Scandinavian Actuarial Journal*, pages 47–55, 1982.

[19] T. Mack. A simple parametric model for rating automobile insurance or estimating ibnr claims reserves. *ASTIN Bulletin*, 21 :93–109, 1991.

[20] T. Mack. Distribution-free calculation of the standard error of chain-ladder reserve estimates. *ASTIN Bulletin*, 15 :133–138, 1993.

[21] T. Mack. The standard error of chain-ladder reserve estimates : Recursive calculation and inclusion of a tail factor. *ASTIN Bulletin*, 29 :361–366, 1993.

[22] P. McCullagh and J.A. Nelder. *Generalized Linear Models*. CRC Press, 1991.

[23] R.J. McDonald, J.B. Butler. Regression models for positive random variables. *Journal of Econometrics*, 43 :227–251, 1990.

[24] M. Merz and M. V Wüthrich. Modelling the claims development result for solvency purposes. *CAS E-Forum*, pages 542–568, 2008.

[25] E. Ohlsson and B. Johansson. *Non-life insurance pricing with Generalized Linear Models*. Springer Verlag, 2010.

[26] C. Pröhl and K. D. Schmidt. Multivariate chain-ladder. In *ASTIN Colloquium*, Zurich, 2005.

[27] G. Quarg and T. Mack. Munich chainladder a reserving method that reduces the gap between ibnr projections based on paid losses and ibnr projections based on incurred losses. *Variances*, 2 :267–299, 2004.

[28] A. E. Renshaw and R. J. Verrall. A stochastic model underlying the chain-ladder technique. *British Actuarial Journal*, 4 :903–923, 1998.

[29] G. Simonet. *Comptabilité des entreprises d'assurance*. L'Argus de l'Assurance, 1998.

[30] C.J. Stone. Additive regression and other nonparametric models. *Annals of Statistics*, 13(2) :689–705, 1985.

[31] R. J. Verrall. An investigation into stochastic claims reserving models and the chain-ladder technique. *Insurance : Mathematics and Economics*, 26 :91–99, 2000.

[32] S.N. Wood. Additive regression and other nonparametric models. *Annals of Statistics*, 62(2) :413–428, 2000.

[33] M. V. Wüthrich and M. Merz. *Stochastic Claims Reserving Methods in Insurance*. Wiley Interscience, 2008.

[34] B. Zehnwirth. *Interactive claims reserving forecasting system (ICRFS)*. Benhar Nominees Pty Ltd. Tarramurra N.S.W., Australia., 1985.